SECOND EDITION

INFORMATION TECHNOLOGY

AN INTRODUCTION FOR TODAY'S DIGITAL WORLD

Richard Fox

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Information Technology

Taylor & Francis

Taylor & Francis Group

http://taylorandfrancis.com

Information Technology

An Introduction for Today's Digital World

Second Edition

Richard Fox

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

Second edition published 2021
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

© 2021 Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, LLC

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Fox, Richard, 1964- author.
Title: Information technology : an introduction for today's digital world / Richard Fox.
Description: Second edition. | Boca Raton : CRC Press, 2020. | Includes bibliographical references and index.
Identifiers: LCCN 2020014295 | ISBN 9780367507206 (hardback) | ISBN 9780367820213 (paperback) | ISBN 9781003050971 (ebook)
Subjects: LCSH: Information technology.
Classification: LCC T58.5 .F695 2020 | DDC 004--dc23
LC record available at https://lccn.loc.gov/2020014295

ISBN: 978-0-367-50720-6 (hbk)
ISBN: 978-0-367-82021-3 (pbk)
ISBN: 978-1-003-05097-1 (ebk)

Typeset in Times
by Lumina Datamatics Limited

For eResources, visit: www.crcpress.com/9780367820213: [insert comp website/eResources URL]

I dedicate this book to the three women in my life: Cheri Klink, Sherre Kozloff, and Laura Smith. I love you all.

Taylor & Francis

Taylor & Francis Group

http://taylorandfrancis.com

Contents

Preface xiii
Acknowledgments and Contributions xv
Author xvii
How to Use This Textbook xix

Chapter 1 Introduction to Information Technology 1

1.1 Defining IT 1
1.2 Computer Systems 6
1.3 Careers in IT 8
1.3.1 Computer Security Specialist 10
1.3.2 Database Administrator 10
1.3.3 Network Administrator 12
1.3.4 Software Developer 13
1.3.5 System Administrator 13
1.3.6 Webserver Administrator 14
1.3.7 Web Developer 15
1.3.8 Overall Career Outlook 16
1.4 A Career in IT 17
1.4.1 Continuing Education 17
1.4.2 Computer Ethics 20
1.4.3 Other Social Considerations 23

Chapter 2 The CPU and Memory 29

2.1 An Overview of a Computer System 29
2.2 The CPU 34
2.2.1 Registers 34
2.2.2 The ALU 36
2.2.3 The Control Unit 38
2.2.4 The Bus 39
2.3 The Fetch-Execute Cycle 41
2.3.1 Instruction Fetching and Decoding 41
2.3.2 Operand Fetching and Instruction Execution 43
2.3.3 The System Clock 45
2.4 The Memory Hierarchy 47
2.4.1 The Impact of Memory Access 47
2.4.2 Types of Memory 48
2.4.3 How the Memory Hierarchy Works 51
2.5 Determining Computer System Efficiency 55
2.5.1 Instruction-Level Pipelining 56
2.5.2 Pipeline Stalls 59
2.5.3 RISC versus CISC 62
2.5.4 Memory Performance 64
2.5.5 Putting It All Together: Benchmark Programs 66

2.6 The Evolution of Computers 67
2.6.1 Before the Generations 67
2.6.2 The First Generation 71
2.6.3 The Second and Third Generations 74
2.6.4 The Fourth Generation 75
2.6.5 The Impact of the Microprocessor 78

Chapter 3 Storage and I/O 79

3.1 Bits, Bytes, Words, and Other Terms 79
3.2 I/O Devices 82
3.2.1 Input Devices 82
3.2.2 Output Devices 87
3.2.3 I/O Devices 92
3.3 Storage Devices 94
3.3.1 Magnetic Disk Storage 94
3.3.2 RAID Technology 100
3.3.3 Optical Disc Storage 103
3.3.4 Solid-State Storage 105
3.3.5 Magnetic Tape Storage 107
3.4 The I/O Subsystem 110
3.4.1 I/O Addressing 110
3.4.2 I/O Modules and Forms of I/O Communication 111
3.4.3 Interrupts 113
3.4.4 Device Drivers 115
3.5 The Study of User Interfaces 115
3.5.1 Health-Related Issues 115
3.5.2 Ergonomics 117
3.5.3 User Interfaces 118

Chapter 4 The Binary Numbering System 123

4.1 Numbering Systems 123
4.2 The Binary Numbering System 126
4.2.1 Bits and Storage Sizes 126
4.2.2 Converting from Binary to Decimal 127
4.2.3 Converting from Decimal to Binary Using Division 128
4.2.4 Converting Decimal to Binary Using Subtraction 129
4.2.5 Conversions Requiring More Than 8 Bits 130
4.3 Octal and Hexadecimal Representations 132
4.4 Negative Numbers and Fractions in Binary 134
4.4.1 Two's Complement for Negative Numbers 134
4.4.2 Fractional Representations 136
4.5 Character Representations 140
4.6 Binary Operations 144
4.6.1 Boolean Operations 144
4.6.2 Binary Addition 145
4.6.3 Other Binary Operations 148
4.7 Error Detection and Correction 151
4.7.1 The Parity Bit 151
4.7.2 More Than One Parity Bit 153

4.7.3 The Parity Byte ... 153
4.7.4 Beyond Parity ... 155

Chapter 5 The Operating System ... 159

5.1 What Is an Operating System? ... 159
5.2 The OS Kernel ... 161
5.2.1 The User Interface ... 162
5.2.2 Process Management ... 163
5.2.3 Memory Management ... 168
5.2.4 Resource Management ... 171
5.2.5 Protection and Security ... 173
5.2.6 Interrupt Handling ... 174
5.3 Forms of Process Execution ... 175
5.3.1 Single Tasking ... 175
5.3.2 Batch Processing ... 176
5.3.3 Concurrent Processing and Context Switches ... 177
5.3.4 Queues ... 178
5.3.5 Multiprogramming ... 178
5.3.6 Multitasking ... 179
5.3.7 Multithreading ... 180
5.3.8 Multiprocessing ... 181
5.3.9 Comparing Forms of Process Execution ... 182
5.4 Other System Software ... 183
5.4.1 Device Drivers ... 183
5.4.2 OS Services ... 184
5.4.3 OS Utilities ... 186
5.5 Booting and System Initialization ... 189
5.6 User Accounts, Groups, and Permissions ... 190
5.6.1 User Accounts ... 190
5.6.2 Permissions and Groups ... 191

Chapter 6 Computer Networks ... 193

6.1 Network Terminology and Concepts ... 193
6.2 Computer Network Hardware ... 196
6.2.1 Wired Connections ... 197
6.2.2 Wireless Communication ... 197
6.2.3 Ethernet and Forms of Cable ... 198
6.2.4 Devices to Connect Networks Together ... 199
6.3 Classifying Networks by Topology and Size ... 205
6.3.1 Network Topologies ... 205
6.3.2 Classifying Network Types by Size ... 208
6.4 Network Protocols ... 210
6.4.1 Ethernet ... 211
6.4.2 The OSI Model ... 212
6.4.3 TCP/IP ... 215
6.5 The Internet ... 222
6.5.1 The Internet Backbone ... 223
6.5.2 The Domain Name System ... 224
6.5.3 The World Wide Web ... 228

6.6 The Internet Today 232
6.6.1 The Cloud 232
6.6.2 IoT 234
6.6.3 The Semantic Web 235
6.6.4 Net Neutrality 237

Chapter 7 Software 239

7.1 Equivalence of Hardware and Software 239
7.2 Categorizing Software 241
7.2.1 Software Releases 241
7.2.2 Categorizing Software by Type 245
7.2.3 Categorizing Software by License 246
7.3 Software Management 249
7.4 Servers 252
7.4.1 Types of Servers 252
7.4.2 Webservers and Apache 254
7.4.3 Proxy Servers and Squid 259
7.5 A Look at Applications Software 262
7.5.1 Word Processors 262
7.5.2 Text Editors 263
7.5.3 Spreadsheets 271
7.5.4 Database Management Systems and the Structured Query Language 275
7.5.5 Other Business Software 279

Chapter 8 Programming and Programming Languages 285

8.1 An Overview of Programming 285
8.2 An Examination of High-Level Programming Languages 290
8.2.1 Early High-Level Languages 290
8.2.2 Structured Programming Languages 295
8.2.3 Object-Oriented Programming 295
8.2.4 Just-In-Time Compilation 301
8.2.5 Other Notable Languages 302
8.3 Types of Instructions 304
8.3.1 Variables and Declarations 304
8.3.2 Input and Output Instructions 305
8.3.3 Assignment Statements 307
8.3.4 Selection Statements 309
8.3.5 Iteration Statements 313
8.3.6 Subroutines and Subroutine Calls 316
8.4 Data Types and Data Structures 319
8.5 Scripting and Scripting Languages 322

Chapter 9 Information Security 329

9.1 What Are Data and Information? 329
9.2 Information Assurance and Security 331
9.3 Risk Management 334
9.3.1 Risk Assessment 334
9.3.2 Disaster Recovery Planning 337

9.4 IT Threats and Solutions 339
9.4.1 Threats That Target Users 339
9.4.2 Threats That Target Software 341
9.4.3 Threats That Target Network Hardware and Networked Computers 344
9.4.4 Unauthorized Access and Password Cracking 347
9.4.5 Malware 349
9.4.6 Backups as a Solution 352
9.5 Encryption Technologies 354
9.5.1 What Is Encryption? 354
9.5.2 Encryption Algorithms 356
9.5.3 Encryption Software 360
9.6 Information and Laws 364

Chapter 10 The Windows Operating System 367

10.1 A History of Windows 367
10.2 Windows File System 371
10.2.1 Windows Default Directories 371
10.2.2 Windows File Types 373
10.2.3 Windows Files and Filenames 375
10.2.4 Windows GUI Tools for Interacting with the File System 376
10.2.5 Windows File Permissions and Sharing Options 379
10.3 Windows Settings 381
10.4 Process Management 389
10.4.1 Windows Processes 390
10.4.2 The Windows Task Manager 392
10.4.3 Windows Services 394
10.4.4 System Performance Monitoring 397
10.4.5 Logging Off, Shutting Down, Booting, Rebooting, and Safe Mode Booting 400
10.5 Windows Administration 401
10.5.1 User Accounts 403
10.5.2 Computer Management Tools 406
10.5.3 The Windows Firewall 410
10.5.4 The Windows Registry 412
10.5.5 Other Administrative Tools 415

Chapter 11 Windows Scripting 419

11.1 Introduction to MS-DOS and PowerShell 419
11.2 MS-DOS Commands 421
11.3 DOS Scripting 429
11.3.1 Variables, Assignments, Input, and Output 429
11.3.2 Conditions and the DOS If Statement 432
11.3.3 DOS for Loop 434
11.3.4 DOS Choice Command 435
11.3.5 DOS Script Examples 436
11.4 Introduction to PowerShell Concepts and Commands 438
11.4.1 Objects in PowerShell 438
11.4.2 A Look at Some PowerShell Commands 439
11.4.3 Help, Aliases, and Entering Commands 442

11.5 PowerShell as a Language 445
11.5.1 Variables and Assignments 445
11.5.2 If Statements 448
11.5.3 Loops 449
11.6 Example Administrative PowerShell Scripts 453

Chapter 12 Unix and Linux 459

12.1 The Development of Unix and Linux 459
12.2 Differences between Unix and Various Linux Distributions 464
12.2.1 Comparing Unix and Linux 464
12.2.2 Differences between Linux Distributions 465
12.3 Examining Linux: Linux Files 471
12.3.1 The Top-Level Directory Structure 471
12.3.2 Linux File Types 477
12.3.3 Inodes and Partitions 480
12.4 Linux Users, Processes, and Services 484
12.4.1 Linux Accounts 484
12.4.2 Linux Processes 487
12.4.3 Linux Services 489
12.5 Why Is Linux Important? 497

Chapter 13 The Linux Command Line 501

13.1 An Introduction to the Linux Command Line and Bash 501
13.1.1 Your Prompt and Entering Commands 502
13.1.2 Forms of Help 503
13.1.3 Wildcards 503
13.1.4 Tab Completion, History, and Command-Line Editing 505
13.1.5 Aliases and Environment Variables 507
13.1.6 Using ~ and Redirection 508
13.2 Linux File System Commands 510
13.2.1 cd (Change Directory), ls (List), and pwd (Print Working Directory) 510
13.2.2 cp (Copy), mv (Move/Rename), and rm (Remove/Delete) 512
13.2.3 Links 514
13.2.4 Accessing Text Files 515
13.2.5 Permissions and Ownership Commands 517
13.3 Process Management Commands 520
13.4 System Status Commands 526
13.5 Basic System Administration Commands 528
13.6 Advanced System Administration Commands 532
13.6.1 File System Commands 532
13.6.2 Software Installation Commands 534
13.7 Bash Scripting 537
13.7.1 Variables and Assignment Statements 538
13.7.2 Input, Output, and Parameters 539
13.7.3 Conditions and if-then/if-then-else Statements 541
13.7.4 Loops 545
13.7.5 Miscellaneous Commands and Comments 548

Index 551

Preface

Why does this textbook exist? At Northern Kentucky University, we have offered a 4-year bachelor's degree in Computer Information Technology since 2004. This program, like the field, continues to evolve. In 2011, we added a breadth-first introductory course to our program. The course's role is to introduce students to many of the concepts that they would see throughout their IT studies and into their career: Computer organization and hardware, Windows and Linux operating systems, system administration duties, scripting, computer networks, regular expressions, binary numbers, the Bash shell in Linux, DOS, managing processes and services, computer security, and careers in IT.

As I was asked to develop this course, I began looking for a textbook. I realized quickly that this wide range of topics would not be found in any single book. A book that covers IT topics, binary numbers, programming/scripting, networks, Windows and Linux, while also providing detail on IT careers? Forget it. Maybe three books, possibly four. So I wrote my own.

In writing this book, I had hoped to create a "computer literacy text for IT majors." What I discovered after writing it was that the book was too specific to our CIT 130 course and also too detailed to be thought of as a computer literacy book. Also, the first edition became obsolete quickly (the book covered Windows 7; Windows 10 was released before the book even hit the shelves, and Linux moved from Red Hat 6 to Red Hat 7 within a couple of years of its publication).

I was given the opportunity to revise this book to bring it up to date and correct what wasn't working. The changes in this edition are enumerated as follows, but it would probably be easier to list what hasn't changed – very little. I rewrote most of the material.

- Chapter 1 is partially rewritten and now contains content from the first edition's Chapters 1 and 16.
- Chapter 2 is completely rewritten from the first edition's Chapter 2 with new content and better explanations. Included in this chapter is content rewritten from the first edition's Chapter 7 on the history of the computer.
- Chapter 3 is a brand-new chapter that contains some rewritten content originally from Chapters 2 and 5 and new material not covered in the first edition.
- Chapter 4 is the only chapter that remains largely intact from the first edition (in which it was Chapter 3). Some material has been removed, and new content was added to make the chapter more useful from an IT perspective.
- Chapter 5's operating system content has been taken from several chapters of the first edition (Chapters 4, 5, 6, and 11), but it is nearly all completely rewritten, with additional material added to make the chapter more complete.
- Chapter 6 has been mostly rewritten to be clearer, more detailed, and a more up to date version of Chapter 12 of the first edition, but many topics have been elaborated upon.
- Chapter 7 was Chapter 13 in the first edition but has been rewritten and elaborated upon and new material added; some material (on databases and data mining) from the first edition's Chapter 15 was moved into this chapter and content from the first edition's Chapter 9 on emacs, vi, and LaTeX have been moved here.
- Chapter 8 on programming has been rewritten and revised from the first edition's Chapter 14; many of the original examples remain, but additional examples have been added as has coverage of some languages omitted from the first edition.
- Chapter 9, the chapter on information (Chapter 15), is now a chapter on information security, and although this chapter contains much of the same content, most of it has been rewritten and expanded upon, especially the detail on computer security threats. Additionally, the section on encryption has been rewritten for clarity and offers more detail.

- Chapters 10 and 11 now consolidate the Windows material that was scattered throughout the previous edition in Chapters 4, 5, 6, 8, 11, and 14. Both chapters are original with only a few figures reused and some of the DOS examples that came from Chapter 14. Furthermore, Chapter 10 contains all new material on Windows administration and Windows tools, and Chapter 11 contains all new material on PowerShell.
- Chapters 12 and 13 provide more detail on the Linux material that came from Chapters 4, 5, 6, 8, 9, 11, and 14 of the previous edition. Some material from the first edition's Chapter 10 has been relocated to Chapter 13, but most of the material on regular expressions is now available online.

To keep the size of this textbook manageable and the cost down, several portions of this text have been provided as supplements available on the CRC Press website: www.crcpress.com/9780367820213. There are two categories of online material: Available to instructors and available to all.

Available to instructors:
- Test bank and answer key
- Instructor's manual (with answers to chapter review questions)

Available to all:
- Chapter review sections for all chapters
- Lab manual
- Lecture notes (PowerPoint format)
- Supplemental reading material, including an essay on the impact of the computer, regular expressions and Linux programs egrep/sed/awk, how to construct a computer, how image files are stored, and how file compression works
- Glossary of all chapter review terms
- Answers to selected chapter review questions

Acknowledgments and Contributions

This book is not so much of a revision but a massive rewrite of the first edition. I am indebted to all the students of CIT 130 who have taken the course over the years and provided me with great feedback as well as many of the CIT 130 instructors. I would also like to thank several colleagues for feedback either on this revision or on earlier material: Scot Cunningham, Maureen Doyle, Wei Hao, Jim Hughes (RIP), and Emily Taylor. I would also like to thank Randi Cohen from Taylor & Francis Group/CRC Press for her help and support. Finally, I would like to thank the following students who provided me some great feedback on drafts of this book: Noah Birrer, Steven Brownfield, Gisela Garcia, Tyler Monroe, and Rachel Weiner.

Taylor & Francis

Taylor & Francis Group

http://taylorandfrancis.com

Author

Richard Fox is a professor of computer science at Northern Kentucky University (NKU). He regularly teaches courses in both computer science (Artificial Intelligence, Computer Systems, Computer Architecture, Concepts of Programming Languages, and Object-Oriented Programming I and II) and computer information technology (IT Fundamentals, Unix/Linux). Fox, who has been at NKU since 2001, is the current chair of NKU's University Curriculum Committee. Prior to NKU, Fox taught for 9 years at the University of Texas – Pan American (UTPA), now known as University of Texas – Rio Grande Valley. He has received teaching awards from UTPA and from NKU and a service award from NKU. Fox received a PhD in computer and information sciences from The Ohio State University in 1992. He also has an MS in computer and information sciences from Ohio State (1988) and a BS in computer science from the University of Missouri Rolla, now Missouri University of Science and Technology, in 1986. Fox has written two other textbooks published by Taylor & Francis Group/CRC Press in the IT field: *Linux and Operating System Concepts* (2015) and *Internet Infrastructure: Networking, Web Services and Cloud Computing* with Wei Hao (2018). He has also authored or coauthored more than 50 peer-reviewed research articles, primarily in the area of Artificial Intelligence.

Richard Fox grew up in St. Louis, Missouri, and now lives in Cincinnati, Ohio. He is a big science fiction fan and progressive rock fan. As you will see in reading this text, his favorite composer is Frank Zappa.

Taylor & Francis

Taylor & Francis Group

http://taylorandfrancis.com

How to Use This Textbook

The original textbook's chapters largely followed the order covered in the NKU CIT130 course. This version of the book presents the topics in what is hopefully a more logical order and one that more closely resembles what you would find in a typical computer literacy book.

The 13 chapters are ordered beginning with an introduction, followed by hardware and binary, operating system, network, software, programming, and ending with miscellany. In this case, miscellany is information security followed by four chapters on Windows and Unix/Linux. The order that material should be covered will vary by your own need, and most of the chapters are independent of each other. The following list shows the order that should be followed to use this textbook effectively.

- Chapter 1 should be covered first.
- Chapter 2 (CPU and memory) should be covered before Chapter 3 (I/O and storage).
- Chapter 4 (binary) should be covered prior to Chapter 6 (networks).
- Chapter 5 (operating systems) should be covered before Chapters 10 and 12.
- If your course includes lab components, you might want to cover parts of Chapters 10–13 throughout the semester. If so, you should make sure you have covered relevant portions of Chapters 3 and 5 before addressing related material in Chapters 10, 12, and 13.
- Chapter 8 (programming) should be covered before Chapter 11 and the Bash scripting portion of Chapter 13.

The following material can be considered optional:

- Chapter 2, Sections 2.5 (computer system efficiency) and 2.6 (evolution of computers)
- Chapter 3, Sections 3.4.1 and 3.4.2 (I/O subsystem)
- Chapter 4, Sections 4.3 (octal/hexadecimal), 4.4 (negative numbers, fractions), 4.6.2 (binary addition), and 4.6.3 (other binary operations)
- Chapter 7, Sections 7.4 (servers) and 7.5 (application software)
- Chapter 8, Sections 8.2 (examination of high-level programming languages) and 8.4 (data types and data structures)
- Chapter 9, Section 9.6 (laws)
- Chapter 11 (Windows scripting)
- Chapter 13 (Linux command line)

Taylor & Francis

Taylor & Francis Group

http://taylorandfrancis.com

1 Introduction to Information Technology

In this chapter, we introduce the field of information technology (IT), cover definitions, explore the components that make up IT, and look at IT careers. The learning objectives of this chapter are to be able to:

- Define the different uses of the term IT
- Explain the types and roles of the components of a computer system
- Describe and differentiate between types of IT careers
- Describe the skill set required to succeed in IT

1.1 DEFINING IT

Information technology (IT) is an umbrella term. It is used to describe careers that use computers; to describe systems of hardware, software, and data; and to describe the staff of an organization who takes care of computer systems. Let's look at each of these in turn.

The term IT has often been used synonymously with *any* career that uses computer systems. In the 1970s through the 1980s, just about any tech-oriented career was placed under the general label, IT. This included, for instance, programmers, system analysts, mechanical engineers, graphic designers, radiologists, and anyone who used a database. Today, just about every field of endeavor uses computer systems to some extent, from accounting to zoology, and so this use of the term IT is no longer useful. But careers in IT are still broader than what we might think. Table 1.1 provides a list of "IT" careers as found in such popular websites as itcareerfinder.com, allaboutcareers.com, and careerizma.com.

The list in Table 1.1 is far from complete. Careers that are more closely tied to IT as it is used today include forms of computer administration. Computer administration includes system administrators, network administrators, and database administrators. We also add help desk and computer support specialists to this list. The IT platform additionally includes the various forms of computer programmers whether they are considered computer scientists, software engineers, software developers, system developers, mobile app developers, web developers, or user interface developers. There are also network architects, cloud engineers, hardware engineers, and robotics engineers. Another category is security specialists such as IT security engineers, security architects, penetration testers, information security analysts, or forensic computer analysts.

This latter list of IT careers makes up the third definition of IT. That is, this list consists of the people who make up the IT staff (except possibly software developers). In IT, it is common that programming is handled by the administrators, engineers, and security specialists.

By shifting the term IT from anyone who uses a computer to those who administer computers tells us that the IT career is more focused today than it was in the past– specifically, the IT jobs are focusing on creating, connecting, configuring, troubleshooting, and securing computer systems. It is this third definition that we will largely concern ourselves with throughout this textbook.

TABLE 1.1 IT Roles in Several Career Fields

Career	Types of Jobs	IT Uses
Health professions	Health information technician	Manage electronic patient records
	Radiology technician	Operate various imaging devices (e.g., X-ray, MRI, CT scan)
	Clinical systems analyst	Troubleshoot/maintain electronic medical record systems and related hardware
Artist	Graphic designer	Use 2D and 3D drawing and editing tools to generate graphics
	Computer animator	Generate computer animation for film, TV, video games, mobile apps, websites
Design and manufacturing	Engineers, architects, etc.	Use computer-aided design (AutoCAD) tools
Business	Information systems Management	Use database, analytics, data modeling software
		Manage IT staff, design IT architecture/ infrastructure
	Information security officer	Protect organization's intellectual property, sensitive data, processes, etc.
	Digital marking manager	Enact marking and PR plans using social media and other forms of telecommunication

Abbreviations: CT, computed tomography; MRI, magnetic resonance imaging; PR, public relations; 2D, two-dimensional; 3D, three-dimensional.

What about the second usage of IT as described previously? This version of IT describes computer systems and their components. Let's spend a little bit of time focusing on this definition by formally defining a computer system.

Did You Know?

The word *computer* originates from the Latin word *computare* meaning "programmable machine".

A *computer system* is the set of hardware and software that people use for processing. The hardware must consist of a processor, if nothing else. The processor *processes* software, meaning that it runs or *executes* programs. We usually refer to programs as *software*. The programs instruct the processor what to do instruction-by-instruction.

People who use computers are known as *end users* (or just *users*). They are also components within computer systems. Without a person, there is nothing for the computer to do. It is the person that decides when to run a program and how to run that program (e.g., what input to use). Although we can automate when programs run, it is the user who sets this up.

Computer systems often contain other forms of hardware: input devices, output devices, telecommunications devices, and storage devices. These four terms overlap as, for instance, storage devices and telecommunication devices allow the computer to obtain input and can be used to send or store output. In all, a computer system performs what we call the *IPOS cycle*: Input–Processing–Output–Storage. This is shown in Figure 1.1. Not all programs will use all four of these functions,

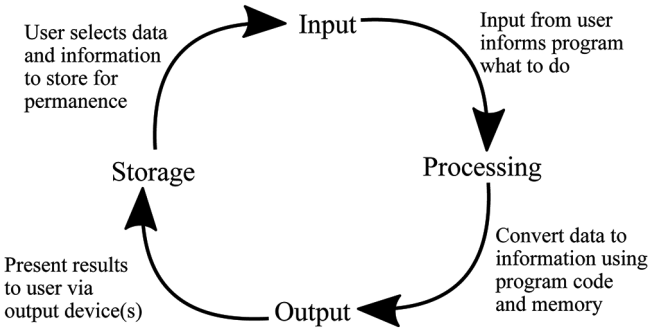
Input
Input from user informs program what to do
Processing
Convert data to information using program code and memory
Output
Present results to user via output device(s)
Storage
User selects data and information to store for permanence

FIGURE 1.1 The input, processing, output, and storage (IPOS) cycle.

but all four are common, and any computer must be able to perform all four. Let's examine each of these four steps.

Input is the process of obtaining information, data, or commands from outside of the computer and bringing it into the computer. Here, "outside" and "inside" are considered the main components of the computer: processor and memory (see the next paragraph). Input devices include keyboard and mouse but also microphones, scanners, sensors, touchpads and touchpoints, joysticks, and others. A telecommunications device can deliver input to the computer as can a storage device.

Processing means to execute or run programs. The program itself is a list of instructions. Each instruction is stored in memory. Processing involves fetching the next instruction from memory, retrieving any data that the instruction will execute on, and executing the instruction on the data. There are several types of instructions: data movement, arithmetic or logic operation, branch, and input/output. Processing is handled by the processor, also known as the *central processing unit* (CPU), and memory.

HUMAN COMPUTERS?

Have you seen the movie *Hidden Figures*? NASA employed many African American women as computers. It was these women's jobs to perform intensive calculations to double-check rocket trajectories. Why did NASA employ human computers?

In fact, the use of human computer dates back to 1935 when the National Advisory Committee for Aeronautics, NACA (NASA's predecessor) hired women. African American women with math skills happened to be available in large numbers, and President Franklin D. Roosevelt had recently enacted legislature requiring that all federal agencies employ people irrespective of race or gender.

Output is the process of sending results of some operation(s) (or of the entire program) and delivering the results to the "outside world". The output should also be transformed into some useful representation that humans can interpret. Output could be text, images (graphics), sound, video, or some combination. Common output devices are the monitor and printer but also include speakers, braille reader, and projector.

Storage is the permanent (or at least long-term) saving of files. This differs from memory, which is used in the short-term and is more limited in storage capacity. The items stored are output from programs, data files used for processing, or computer programs. Our most common form of storage is the hard disk drive. We also use flash memory in the form of solid-state drives and USB drives and although somewhat outdated, both optical disc and magnetic tape as well.

Computers used to be clearly recognizable. They were large machines stored in separate rooms (kept clean and most likely cooled by a powerful air conditioner). In the 1970s, personal computers (PCs) were first marketed, leading to two sizes of computers: large computers that could run dozens, hundreds, or even thousands of programs and personal computers that would run one program at a time. The PCs were small enough to fit on a desktop, whereas the larger computers were the size of one or more filing cabinets and, in some cases, would fill an entire room.

Today, we have an even greater variety of computers as we have supercomputers that are as large as the computers from the 1940s but are far more powerful, and we have computers that are quite literally the size of a grain of rice (the smallest computer as of 2019 is just 0.3 mm in size). The most common type of computer today is not the PC but the smartphone. According to techjury.net, as of March 2019, there were more than 2.71 billion smartphones in use. Statcounter.com estimates that there are a little more than 2 billion PCs and laptops and 145 million tablets (as of 2018).

Table 1.2 provides a comparison of the types of computers currently available. The table illustrates for each type its usage, cost, size, processing capability, and size and type of memory. Notice that the smaller the computer, the less likely it will contain both memory and storage.

Figure 1.2 illustrates several types of computers we have today. We see an all-in-one desktop computer in the upper left and a desktop with separate system unit (tower) in the upper middle. On their right is a laptop with the monitor, keyboard, and system unit combined. In the middle row there are mobile devices, including a book reader on the right and at the bottom is a music player. The book reader and music player may be able to do more than just display books or play music, respectively, but are more limited and so might be called *special-purpose* computers rather than *general-purpose* computers.

TABLE 1.2 Types of Computers (see Table 3.1 for Storage Capacity Definitions)

Type	Typical Use	Size	Cost (US$)	Processing Capability	Memory and Storage
Supercomputer	Scientific/research computing; uses massive parallel processing	Thousands of square feet	10–100 millions	Thousands to millions of processor cores; trillions to quadrillions of operations per second	Main memory up to 100 GBs and disk storage in the PB range
Mainframe computers	Organization-level processing such as bulk data processing	The size of 1 or a few storage cabinets	10–100,000	Hundreds to thousands of millions of operations per second	10 GBs of memory, TBs or more of storage
Minicomputers	Organization-level processing for smaller organizations; minicomputers have largely been phased out with organizations using servers instead	1 storage cabinet	A few to 10,000	Tens to hundreds of millions of operations per second	A few 10 GBs of memory, TBs of storage
Servers	Large-scale storage of files and processing to handle file requests; there are many types of servers including web servers, database servers, file servers, and e-mail servers	1 or more storage cabinets depending on the expected load of the server	1000–10,000	Millions of operations per second	A few to 10 GBs of memory, TBs to PBs of storage
Workstations	High-performance PCs; phased out once PCs were capable of multitasking	Same as PCs or slightly larger	5–20,000	Millions of operations per second	A few to 10s of GBs of memory, 1 to a few TBs of storage

(*Continued*)

TABLE 1.2 (*Continued*) Types of Computers (see Table 3.1 for Storage Capacity Definitions)

Type	Typical Use	Size	Cost (US$)	Processing Capability	Memory and Storage
PCs/laptops	Small processing jobs, Internet access, multimedia/gaming purposes	2′ tower or smaller with peripheral devices (monitor, keyboard), laptops no more than 2′ × 1′ × 3″	500–1200	Millions of operations per second	8–16 GBs of memory, up to 1.5 TB of disk storage
Mobile devices (tablets, smartphones)	Telephone/texting, small-scale applications, Internet access	Handheld size	100–700	Millions of operations per second	4–8 GB of memory (at most, external storage only)

Abbreviations: GB, gigabyte; TB, terabyte; PB, petabyte; PCs, personal computers.

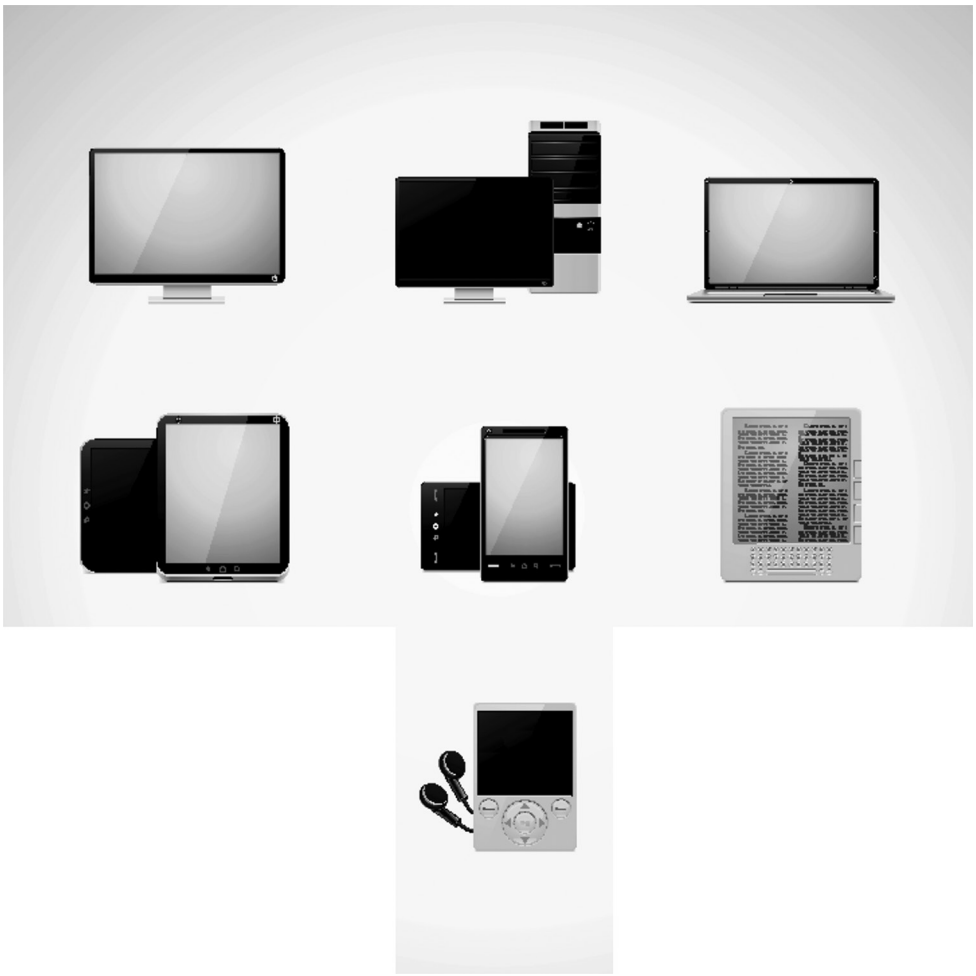

FIGURE 1.2 A comparison of computer types.

The special-purpose computer is built specifically to solve a particular type of problem, and so either cannot or does not run programs to solve other types of problems. Early computers (from the 1930s and 1940s) were special purpose. A general-purpose computer is one that can run a wide variety of programs – virtually any program that we know how to write. We have only recently, with the advent of mobile devices (like mp3 players, e-book readers, GPS systems), begun to see a rise in special-purpose computers.

So, there you have it, the three different uses of the term IT. Over time, the third usage has become the more predominant version in our society but still not the only one in use. As we move forward in this text, we will largely concentrate on the latter two uses: computer systems and their related components and the roles and duties of IT personnel.

SECTION 1.1 ACTIVITY

1. Look around your house, and identify every device that contains either a processor, memory, or both.
2. How many computers do you own? For the purpose of this activity, we will assume any device that can be programmed is a computer. How many general-purpose computers do you own?
3. Which of the following do you use at home: Wi-Fi, social-media sites, cloud storage, free e-mail account(s), 5 G cell networks? These are all recent (last 20 or 25 years) innovations in technology.
4. If you work, identify your IT personnel. If you are in school, do the same. How many IT personnel are there, and do they each have specific roles or do they all share the same general job?
5. Of the types of computers listed in Table 1.2, which types have you used? Which types has anyone in your family used?

1.2 COMPUTER SYSTEMS

In this section, we concentrate on the components of a computer system. We offer definitions and examples throughout. This section is to familiarize you with these components. Each is covered in more detail in Chapters 2 (CPU and memory) and 3 (I/O and storage).

The *processor* (CPU) is the main component of any computer. Its role is to execute program instructions. If we can solve a problem by writing a program, our general-purpose computers can run that program.

Processors are constructed to execute a specific list of instructions known as the processor's *instruction set*. Collectively, the instructions and the rules for their usage are known as the *machine language* of the processor (also sometimes referred to as the processor's native language). Unfortunately, machine language is extremely cryptic and hard to use. By the late 1950s, programmers had written programs to translate code from a slightly more comprehensible language, *assembly language*, into machine language. These programs were called *assemblers*. But assembly languages are also challenging to use, especially when writing a program of more than a few dozen instructions.

By the end of the 1950s, *high-level languages* were being introduced and along with them, *compilers* to translate program code from a specific high-level language into a specific machine language. See Figure 1.3 for a comparison between a single C instruction, its equivalent assembly code, and its equivalent machine code for an Intel processor.

There are two things to notice in Figure 1.3. First, a single instruction in C maps to several assembly and machine instructions, whereas each assembly instruction equates to exactly one machine instruction. Second, although hard to understand, there are at least some aspects of the assembly code that might be comprehensible such as mov ("move") and [a], [b] and [c]. The machine code, on the other hand, looks like nonsense. In the figure, the machine code is represented in hexadecimal

```
C code:              Assembly code:              Machine code:

c = a * b + b;       mov eax,dword ptr [a]       8B 45 F8
                     mov ecx,dword ptr [b]       8B 4D EC
                     lea edx,[ecx+eax*2]         8D 14 41
                     mov dword ptr [c],edx       89 55 E0
```

FIGURE 1.3 Comparing high-level to low-level code.

notation but would be stored in the computer in binary, and so that first instruction would actually look like 10001011010000101111111000! We explore programming languages in Chapter 8.

The flexibility of our processors makes them unique among *all* of our human artifacts. The processor is capable of solving a myriad type of problems. Other human-made devices are not nearly as impactful. Cars transport objects. Books convey words and pictures. Microwave ovens heat food. Fans blow air. But computers, because of their processors, do a great many things. As mentioned, a computer can solve any problem so long as we know how to write a program to solve it.

We focus on the processor in Chapter 2. For now, we will just leave it as described: a device that executes program instructions that are written in the processor's instruction set. To execute program instructions more efficiently, we equip our computers with forms of storage. There are numerous forms of storage, but we break them into two distinct categories: short-term storage in the form of various types of memory and long-term storage.

Most processors contain internal storage to store information about the current instruction. These storage locations are known as *registers*. One register will store the instruction itself. Another register will store the location in memory of where to retrieve the next instruction. Yet other registers will store data that this instruction is to operate on or recently executed instructions have used or the results of some of these instructions.

Registers are for immediate usage. That is, the registers contain information that the processor is using right now, has used recently, or will use in the immediate future. Other information is stored in short-term storage that we refer to as *memory*. There are many forms of memory.

Cache memory is often stored in the processor. *Main memory* is stored separately from the processor, usually on chips inserted into the computer's motherboard. There are several differences between cache and main memory. First, cache memory is faster technology and so responds more rapidly. Second, cache memory is both bulkier and more expensive. Because of these differences, we usually have far less cache memory than main memory in our computers. Third, as noted, cache memory is usually stored on the processor itself (although there might also be cache memory stored outside of the processor), whereas main memory is always stored outside of the processor. Finally, when the processor needs something from memory (e.g., instruction, data), it looks for it in cache first.

The most recently used items will be found in cache, and less recently used items should be found in main memory. The CPU then must obtain the item it is looking for by searching each form of storage until it locates what it is looking for. The collection of storage - registers, cache, main memory and virtual memory - all make up what we call the *memory hierarchy*. We will explore the memory hierarchy, including these various forms of memory, in Chapter 2.

Recall the IPOS cycle (see Figure 1.1), input, processing, output, and storage. Both the processor and the memory hierarchy are used during the processing step. Storage conveys long-term, or permanent, storage. For this, we use some form of storage device. Until recently, storage was always placed on a medium that a storage device would access. Such media include magnetic disk, optical disc, and magnetic tape. A more recent form of storage is *called flash memory*, which we use to build both solid-state disk drives and USB drives. We use long-term storage to store our files. Files can be data files that we will use as input to a program, processed results from running programs, or programs themselves.

Input devices receive input, usually from a human, and bring that input to the processor to process. Input could be moved into a register or memory. Output devices receive processed information

from the computer and send that information to a device for human viewing. Although input and output devices were originally limited to magnetic tape (storage, which was used for both input and output) or punch card reader (input) and printer (output), today we have dozens of types of input and output devices. Some are highly specialized to make interaction with the computer easier or to fit a specific application. For instance, a data glove can sense hand movements and be used to control a virtual-reality program or computer-aided drafting (CAD) software or be used to input sign language. Some devices can be used for both input and output such as a touch screen. We look at I/O (input/output) devices and storage devices in Chapter 3.

One additional form of computer system device is a network. The network is not part of a computer, but the computer can connect to a network. This links the computer to other computers as well as I/O and storage devices. Data coming from the network can serve as input and data going across the network can serve as output. We explore computer networks and network devices in Chapter 6.

As we noted previously, a computer system also comprises the software run and the users who use the computer. Software is divided into two types: system software and applications software. System software is the operating system and supporting programs. We explore the operating system in Chapter 5 and focus on Windows and Linux in detail in Chapters 10 through 13. Application software makes up the programs that users want to run. We explore applications software in Chapter 7.

SECTION 1.2 ACTIVITY

1. How much main memory does your computer have? If you cannot locate this as a piece of information that came with the computer (via a manual or sticker on your computer or on the box), see if you can identify the amount from the computer itself. For instance, in Windows, open up the Control Panel and select System.
2. Similarly, try to identify your processor type, clock speed, and number of cores.
3. What are the input devices you use in your computer (or smartphone)? What are the output devices you use?

1.3 CAREERS IN IT

We focus our look at IT careers on those specific to the development of and administration of computer systems. Up until recently, IT staffs were more often people who were technically savvy but perhaps lacked a formal IT education. IT personnel might be drawn from people with a wide variety of backgrounds or perhaps selected because they had studied computer science and so were the deemed the "computer experts". It was not uncommon for the manager to tap a person to install software, set up a computer, or even configure the network because that person had done something similar at home as a hobby. This led to that person being assigned as the de facto IT person for the organization. Today, there are 2- and 4-year IT degree programs as well as graduate degrees in IT. The IT staffs are being populated as much from this pool of graduates as from anywhere else.

The word "IT" remains as an umbrella term. There are many different types of activities that are tasked to the IT staff. They all revolve around installing, configuring, programming, securing, and troubleshooting the IT infrastructure. The IT infrastructure will include hardware, software, and networks. The hardware might include servers, and the software may include specialized server software (e.g., a web server). The job of the IT staff may also include producing documentation for the employees (e.g., how to log in to your account and change your password) and generating IT policies and procedures for management.

Table 1.3 describes some of the current types of IT careers. The table lists types of operations that each IT role takes on. One word that does not appear in Table 1.3 but is common across all

TABLE 1.3 IT Careers

Role	Types of Tasks	Types of Knowledge
Computer security specialist	• Secure computer systems • Secure computer networks • Analyze break-ins • Find and fix security holes in operating systems and applications software • Develop and employ algorithms for encryption • Test operating systems and software for security holes • Perform forensic analysis and help build legal cases	• Common forms of security breaches/attacks • TCP/IP • Operating systems • Programming • Security tools • Penetration testing tools • How to read and interpret log files • Legal procedures
Database administrator	• Install and configure database management system, possibly over a network and in conjunction with a front-end web server • Secure the database • Create accounts • Monitor and troubleshoot database access, maintain efficiency of access	• Database management systems (e.g., Oracle) • Relational and nonrelational databases • SQL • Database design • Networks and network communication, client-server models • Storage technologies
Network administrator	• Install and configure network media (cables, broadcast devices) • Secure network (firewalls, other security devices, spam filters) • Install and configure network server software (e.g., DHCP server, DNS server)	• TCP/IP and the OSI model • Routers and switches • Ethernet technologies • Wireless access points • Network and Wi-Fi encryption technologies
Software developer	• Software engineering • Application development • Operating system development • Front-end development • Back-end development • Mobile-app development	• High level languages (e.g., C++, C#, Java, Python) • Algorithms and data structures • Knowledge of development environments (IDEs) • Debugging skills
System administrator	• System administration of many computers (account creation, software installation, security) • Operating system troubleshooting • Hardware troubleshooting and installation • Writing how-to manuals	• Operating systems • Scripting • IT policies
Web administrator	• Install and configure webserver • Secure webserver • Install and configure proxy server • Install and configure back-end database	• Webserver software (e.g., Apache, IIS) • HTTP, HTTP2, HTML, CSS • TCP/IP • Web development (possibly)
Web developer	• Front-end web page development • Server-side scripting • Designing web pages	• HTML, CSS • JavaScript • Php and/or Perl or related languages • HTTP/HTTP2

IT careers is *troubleshooting.* Troubleshooting is a form of problem solving that attempts to either repair something that is not working or improve the effectiveness of something that is not working well. In any form of IT career, troubleshooting is required to figure out why some aspect of the computer system failed or how the computer system can be improved. There are different forms of troubleshooting depending on whether the specific role is of a system administrator, network administrator, programmer, security specialist, or other.

Let's now explore each of the careers from Table 1.3 in more detail. The following subsections describe each type, providing such detail as tasks, salaries, chances for career advancement, and certifications that might help achieve such advancement.

1.3.1 Computer Security Specialist

Computer security specialist is listed in Table 1.3 as a single career, but there are, in fact, many different types of computer security jobs, including cryptographer, security analyst, security architect, security consultant, security engineer, and secure software developer. We do not differentiate between these, although they call for different areas of expertise (for instance, the cryptographer will require a good deal of mathematical knowledge, and the secure software developer will need to be well-versed in software engineering). The security specialist itself is often considered an entry-level security job. But we will continue to use the term in this subsection to represent more advanced positions such as cryptographer, security manager, security architect, chief information security officer, security analyst, security engineer, and penetration tester.

A computer security specialist must understand computer systems. This means a depth of knowledge in both operating systems and hardware. The specialist must also understand the nature of attacks that can occur both on computers and computer networks. Thus, the security specialist must also understand network protocols, firewalls, and the hardware that connects network components together. The computer security specialist may also be responsible for securing servers and would have to understand the specific type of server software such as a web server or a DNS name server.

Among the skills that the security specialist must know are how to use the variety of tools available, how to write secure code, and how to document discovered vulnerabilities. Tools include those to detect break-ins, perform penetration testing, or otherwise search for vulnerabilities in a computer system. Understanding encryption is a must, whether that is how to use the available encryption tools or a deeper understanding of the mathematics behind cryptography. Legal knowledge may also be useful if the role is to document break-ins for court cases.

All areas of IT require continued education, but it is perhaps most critical for computer security specialists to maintain their education. The reason is threefold. First, this field changes as fast as or faster than any other. As soon as a solution to a type of attack is found, attackers derive new approaches. Second, cyberattacks cost our society billions of dollars annually, and so it is extremely important to keep them secure. Third, there are plenty of methods to continue one's education through certifications, and the security field has as many or more than other fields. Some significant security certifications are CompTIA Security+, Certified Information Systems Security Professional, Certified Ethical Hacker, Certified Information Security Manager, and GIAC Security Essentials.

Currently, computer security careers can pay anywhere from $49,000 to more than $200,000 per year. The average salary is low $70,000s. The chief information security officer, applications security engineer, and cybersecurity analyst are among the highest salaries within the computer security field.

1.3.2 Database Administrator

The *database administrator* has a long list of important responsibilities, but the focus revolves around one piece of software: the organization's database management system (DBMS) and the databases created by and manipulated from the DBMS. Tasks will include installation, maintenance, performance analysis, and troubleshooting of the DBMS. There are specific applications, performance tuning parameters, account creation, and procedures associated with the DBMS that are generally not found with other forms of software. The database administrator may also be involved with the integration of data from older systems to newer ones. Since as many database systems are now available over the Internet (or at least over local area networks), the administrator must also ensure proper security measures for remote access.

There are many different types of DBMSs. The most popular are relational databases such as Oracle, MySQL, or Microsoft SQL Server. But other types of database systems are on the rise, including NoSQL databases and cloud-based databases. As with all areas of technology, the continued growth of new database technology requires database administrators to be in a pattern of constant learning and self-improvement.

Unlike a webserver administrator who will probably not be involved in web development, a database administrator is often expected to play a role in application design and development. This may be by assisting the application developers with the design of the databases needed to effectively support those applications. This requires a unique set of skills that include database modeling, schema development, normalization, and performance tuning.

Because of the more specialized nature of the database administrator, salaries are higher than in the other IT roles listed here. Median average salaries are around $87,000 with the highest salaries being more than $112,000. Certifications are primarily those of particular DBMS software, such as the Microsoft Certified Database Administrator, Oracle Certified, and MySQL Certification.

SOFT SKILLS

Employers are often as interested in your soft skills as your hard skills. Hard skills are typically learned through coursework, but soft skills are learned through experience or are already part of your personality. Some of the soft skills that employers seek include:

- Able to learn from criticism
- Able to network (socially)
- Able to write and document work
- Articulate thoughts and ideas
- Can handle pressure and stress
- Has common sense
- Is confident
- Is courteous of others
- Uses critical thinking
- Has a desire to learn
- Is flexible, adaptable, and trainable
- Has good listening skills
- Has good personal appearance
- Has integrity
- Is honest
- Is innovative
- Demonstrates leadership capabilities
- Is logical
- Makes eye contact
- Is organized
- Has a positive attitude
- Relates to coworkers
- Is reliable
- Is resourceful
- Can self-supervise
- Has a strong work ethic and is hard working
- Is a team player
- Uses time management skills.

1.3.3 Network Administrator

Network administrators are responsible for the installation and maintenance of hardware and software that make up a computer network. Network hardware includes specialized network devices such as switches, routers, hubs, firewalls, adaptive security appliances, wireless access points, and voice-over-IP (VoIP) phones. Network software is often built into operating systems installed on those hardware devices, the protocols used to communicate over the network, and specific network applications.

The network administrator will need to be familiar with routing protocols, access lists, trunking protocols, spanning tree protocols, virtual LANs, virtual private networks, IPv4, IPv6, and intrusion detection systems. Network administrators may also be tasked with securing the network; this task may be accomplished in conjunction with other IT personnel like system administrators.

If the network slows or shuts down, network administrators may find themselves in a high-pressure troubleshooting role. Administrators must use their skills and knowledge to diagnose the issues, identify potential solutions, and implement the selected solution. A solution should solve the given problem most effectively in terms of user convenience, efficient access, and reasonable cost. Issues and solutions should be well documented.

Higher-level network administration positions, sometimes referred to as *network architects* or *network engineers*, are often responsible for designing networks that meet the performance and capacity needs of an organization. These jobs include roles of choosing appropriate network hardware and planning the allocation of network addresses across an organization.

Network administrators all require a comprehensive knowledge of the TCP/IP protocol stack as well as other network protocols that follow the Open Systems Interconnect (OSI) model. However, network administrators often focus on specialties such as network security, wireless administration, VoIP, management of Internet service providers, and network reliability and capacity planning.

From this list of duties, it is obvious that network administrators require a significant amount of training and experience to be successful in their roles. Because of the changing nature of technology, network administrators must constantly learn new technologies through trade journals, conferences, publications, and continuing education. Various career certifications in these specialty areas are a valuable addition to a network administrator's resume. Among the most sought after certifications are the Cisco Certified Network Associate (CCNA), Cisco Certified Network Professional (CCNP), Certified Novell Engineer (Novell CNE), Certified Wireless Network Administrator (CWNA), and CompTia's Network +.

Salaries for network administrators (as of 2019) range from $46,000 to as high as $104,000 with a median average salary of $68,200. Management of company networks currently pays around several thousand dollars a year more than working for telecommunications carriers, which pays substantially better than similar jobs working for schools of any level.

IT JOB QUALIFICATIONS

- Bachelor's Degree in Information Technology, Computer Science, or related field is required.
- Excellent IT, communication, leadership, and management skills.
- Task-oriented
- Ability to fill multiple roles simultaneously.
- Proven track record of maintaining IT structural integrity.
- Industry certifications and memberships are a plus.

1.3.4 Software Developer

Historically, computer programmers have come from any number of disciplines, including the sciences, mathematics, and business. The computer science degree became popular starting around 1980. Today, companies do not hire programmers, preferring instead to hire graduates of computer science, software engineering, or computer engineering programs. The three programs have a good deal of overlap in terms of courses and learning outcomes, but each has its own area of emphasis. Computer science adds a component on the foundations and theory of computation, software engineering adds user interface and platform-specific content, and computer engineering adds processor design and electrical engineering material.

Be it a computer scientist or software engineer, the typical career path is to become a *software developer*. The developer is tasked with the design, implementation, and debugging involved in software creation. Some developers work all phases of this life cycle, whereas others concentrate on one specific task such as software maintenance or software testing. Developers produce all types of software whether it is applications for desktop or laptop computers, mobile apps, client-side or server-side scripts, operating system software, network software, or some other form.

The computer science discipline provides a different type of foundation than that of IT. For the computer scientist, a mathematical grounding is necessary to study topics in computability and solvability. These topics give the computer scientist the tools to measure how challenging a given problem is to solve (or whether it is solvable at all). A person who studies computer science also studies computer organization, computer architecture, and operating systems to understand the underlying nature of the computer system as well as a wide variety of algorithms, data structures, and programming languages. Other topics might include web development, mobile app development, cloud computing, compiler design, and artificial intelligence.

As with the IT individual, the computer scientist or software engineer must keep up with the field. Whereas the IT specialist will largely study the latest operating systems, their features, tools, and security measures, the computer scientist will study new languages and programming techniques. There are certifications for software developers including the Microsoft Certified Professional Developer (MCPD), Microsoft Certified Solutions Developer (MCSD), Microsoft Certified Professional Programmer, and Oracle Java Certification. However, in most cases, earning a computer science degree will be far more important than receiving certification.

Salaries for developers vary from around $50,000 to $104,000 with an average of more than $70,000. Of all of the careers listed in this section, it is the software developer that is expected to see the most growth in the next decade with an estimated growth rate of 21%. As mentioned, historically software developers came from a variety of backgrounds, including many who never studied computer science. Today, the computer programmer is becoming obsolete because companies instead choose people who have studied computer science or software engineering.

One particularly potent mixture of coursework is to combine the IT degree with computer science coursework. This provides the IT individual with an excellent programming foundation. Alternatively, the computer scientist is well served by taking additional IT courses to gain an improved understanding of operating systems, networks, and computer security.

1.3.5 System Administrator

Whereas network administrators are responsible for the performance of networking hardware and software, *system administrators* are responsible for installing, maintaining, and securing the computer systems themselves. The system administrator's role covers hardware, software, and system configuration. Computers might include mainframes, minicomputers, network servers, and PCs. The system administrator should also be familiar with multiple operating systems because it is likely that the organization has computers of several platforms.

The responsibilities of the system administrator are varied depending on the size of the organization. The system administrator will, at a minimum, be required to install and configure operating systems

on the organization's machines, install new hardware, perform software upgrades and maintenance, handle user account and password administration, perform backups and recovery, and perform system monitoring, tuning, and troubleshooting. The system administrator may also be involved with application development and deployment, system security, and reliability.

The system administrator may also be required to install, configure, maintain, and secure any servers such as a file server or a webserver. These duties may fall on the network administrator, or in the case of a webserver and database server, the tasks may be handled by specific webserver and database administrators. The system administrator would most likely have to perform some programming through shell scripting so that repetitive tasks can be automated. A greater amount of programming might be required depending on the organization's size and staff. Policy creation and IT infrastructure planning may also be tasks assigned to system administrators.

Smaller organizations will hire fewer IT staff, and thus, the system administrator's duties may overlap those of the other administrator roles such as the network administrator. On the other hand, smaller organizations will likely have fewer IT needs and so the system administrator may have more time available to undertake the duties of the network administrator.

The knowledgeable system administrator will turn to any number of web forums available whereby system administrators discuss recent problems and look for solutions. There are also numerous certifications available. These include the Microsoft Certified Systems Administrator (MCSA), Microsoft Certified IT Professional (MCITP), Red Hat Certified Engineer (RHCE), Red Hat Certified Security Specialist (RHCSS), and Novell's Certified Novell Administrator (CNA) as well as the network certifications, which is already mentioned. Salaries for system administrators are in the same basic range as those of network administrators.

1.3.6 Webserver Administrator

The *webserver administrator*, also known as the *webmaster*, is responsible for maintaining websites. This differs from the development of websites (see Web Developer in Subsection 1.3.7). Specifically, the webserver administrator must install, configure, maintain, secure, and troubleshoot the webserver. The webserver is software that runs on one or more computers. The webserver administrator may or may not be responsible for the hardware. The webserver administrator may be part of the system administrator staff and so duties may overlap. If separate from system administrators, then the webserver administrator will have to work with the system administrators when it comes to installation of software, creation of user accounts, and access to the file space that makes up the website.

Configuration involves setting up configuration files to specify such directives as where the web pages will be located, who can access the web pages, whether there is any authentication requirements, what forms of security might be implemented, whether and what documents will be encoded or encrypted, and what scripting languages the webserver may execute. Monitoring the webserver's performance permits the administrator to fine-tune aspects of the server. This might include, for instance, installing larger capacity or additional hard disks, adding file compression, and reducing the number of server side scripts. Additionally, monitoring automatically generated log files provides the administrator with URL requests that could be data mined to discover client browsing behavior.

Securing the webserver becomes critical for business. The webserver administrator may be required to set up password files and establish locations where server scripts can be stored and tested. Security may be implemented directly through the webserver software, added through the operating system, or some combination. Security issues may be discovered by analyzing the same log files that are used for data mining.

Webserver software is made up of various lesser programs. These programs handle any number of server-related tasks from the simple retrieval of files of URL requests to logging to error handling to URL redirections. There are free webservers such as Apache and Nginx as well as popular commercial webservers such as IBM WebSphere, Oracle WebLogic, and Microsoft IIS. The webserver administrator may be involved in the selection of the webserver software, but it is the webserver

administrator's job to understand the software selected and make recommendations on how best to support the software with adequate hardware. Hardware decisions might involve whether the server should be distributed across numerous physical computer servers, whether the server requires multiple IP addresses, and what type of storage system should be used. In monitoring the webserver's performance, the webserver administrator can make informed recommendations on upgrades and improvements to the hardware.

Webserver administrators may be required to either perform application development themselves or to support the developers. Support might be in the form of installation of additional webserver modules such as those that can execute Perl or PHP code. There may also be policy decisions that impact development, such as the requirement that all scripts execute outside the webserver on another computer or all scripts being stored in a common cgi-bin directory.

Although the webserver administration may work for a large organization that wishes to have a presence on the Internet, there are also opportunities to work for companies that host websites. In such a case, a single server might be used to host multiple websites of different companies. The administrator has the added challenges of ensuring effective webserver(s) performance while dealing with several groups of web developers.

Finally, the webserver administrator may also be required to install, configure, maintain, and secure related Internet servers. Specifically, the webserver administrator may be in charge of running a proxy server and/or a DNS name server.

Webserver administration salaries can vary greatly from $40,000 to $90,000 with an average around $55,000. There are fewer certifications available for webserver administrators. The primary form is known as a Certified Internet Webmaster Server Administrator. This is one of many under the abbreviation CIW, but most of the CIWs involve web development.

1.3.7 Web Developer

There are many different roles that one might play in web development. An organization might hire a single individual to get their company "up on the web" or a company may hire a staff of web developers. An organization may employ an external company to accomplish this task. The specific role that a *web developer* plays will then vary depending on the company and its circumstance. What is clear, however, is that the web developer has different skills than the website administrator.

Web developer skills include the ability to directly write web pages using hypertext transfer markup language (HTML) and cascaded style sheets (CSS). The web developer will probably need to know one or more scripting languages. There are two types of web scripting: client-side and server-side. The most commonly used client-side scripting language is JavaScript. Two related technologies are AJAX (a combination of JavaScript and XML) and JQuery (a set of library utilities available in JavaScript). Another common tool in client-side development is ActionScript, which is a tool that generates interactive animation applications for Adobe Flash (although Adobe is planning to end the use of Flash in 2020). Server-side scripting can be accomplished by a variety of languages. Currently, the most popular are PHP, Java, Ruby (specifically Ruby on Rails), Python, ASP.NET, C#, C++, and JavaServer Pages (JSP). Although a web developer would not be expected to know all of these languages, it is likely that the web developer would use several of them.

Programming is an important aspect of web development, but web developers are usually not required to have computer science degrees. Instead, the web developer should also have knowledge of user interface design and user interface issues. This would allow the developer to build aesthetically pleasing web pages with the use of proper color, layout, graphics, animation, and so forth. The web developer would also need to understand many of the technical and societal issues related to the Internet. For instance, understanding the nature of network communication, bandwidth speeds, disk caching, and the use of proxy servers would help a web developer build a more efficient and easier to download website. Understanding societal issues can be critical in securing a website from attack, ensuring data integrity of the website, and ensuring that the website content

is politically correct. There are also a number of accessibility concerns so that people with various types of disabilities can still visit and maneuver the site without complication. It is now a federal law that websites of federal agencies and agencies funded by the government be accessible.

Web development itself includes several different tasks that incorporate graphic design and art, business, database access, telecommunications, human-computer interaction, and programming. Web developer positions are in extreme demand these days. On the other hand, they have a tremendous salary range, possibly reflecting the wide range of companies that are looking to create and maintain web portals. Salaries can be as low as $30,000 with a ceiling of more than $100,000 but an average of around $70,000. There are also a number of certifications available to improve status. These include the Microsoft Certified Professional Developer, several CIW Web Developer certificates (see Section 1.3.6), HTML Developer Certification, and Sun Certified Web Component Developer to name a few.

1.3.8 Overall Career Outlook

No matter which area of IT you might decide to specialize in, nearly all areas are among the fastest-growing careers and offer competitive salaries compared to other careers. Table 1.4 is a set of data assembled from the US Bureau of Labor Statistics (note that webserver administrator is not available from the US Bureau of Labor Statistics and so was compiled from other sources). In the table, you will find many of the careers discussed in this section, along with a few others not mentioned. The table provides employment figures for 2018 (the last year that such data was available at the time of this writing), expected growth through 2028, median salary, and expected level of education. Note that all data are restricted to the United States.

From the table, we see nearly every field is growing. The exception is computer programmer. Although there are a good number of programming jobs available, most organizations prefer to hire a software developer (i.e., someone who has a formal degree).

The US Bureau of Labor Statistics currently anticipates the collective set of computing careers to have the fourth-greatest growth between 2018 and 2028, behind only health fields, personal care/service occupations, and social service occupations. IT careers are anticipated to grow faster

TABLE 1.4 IT Career Outlook According to the US Bureau of Labor Statistics

Career	Number of Jobs in the Field in 2018, in Thousands	Projected Number of Jobs in 2028, in Thousands (% Growth Listed in Parentheses)	Median Annual Salary as of 2019, in $US	Expected Minimum Education/Amount of Experience Expected for Entry-Level Position
Computer engineer	64.4	68.4 (6%)	114,600	Bachelor's/None
Computer network architect	159.3	167.7 (5%)	109,020	Bachelor's/5+
Computer programmer	250.3	232.4 (−7%)	84,280	Bachelor's/None
Computer support specialist	863.1	946.2 (10%)	53,470	Some college (no degree)/None
Computer systems analyst	633.9	689.9 (9%)	88,740	Bachelor's/None
Database administrator	116.9	127.4 (9%)	90,070	Bachelor's/None
Information security analyst	112.3	147.8 (32%)	98,350	Bachelor's/Less than 5 years
Network and system administrator	383.9	412.1 (5%)	82,050	Bachelor's/None
Software developer	1365.5	1649.6 (21%)	105,590	Bachelor's/None
Web developer	160.5	181.4 (13.0%)	69,430	Associate's/None
Webserver administrator	N/A	N/A (20.0% through 2022)	66,130	Associate's or higher/None

than careers in such fields as construction, business/finance, life and social sciences, legal occupations, educational occupations, management, and architecture/engineering. Software developers and information security analysts have the greatest job growth within IT and rank 9th and 17th, respectively, out of more than 1000 different careers.

Note that many of the top 8 careers (those with greater growth rates than any in IT) have median annual wages of less than $40,000. These include, for instance, solar photovoltaic installers, home health aides, and personal care assistants. These fields' growth rates are large in part because the current number of people in such fields is relatively small. There are, for instance, only 11,300 people currently working in solar installation, whereas the demand in 10 years is expected to be more than 23,000 jobs.

SECTION 1.3 ACTIVITY

1. IT degrees are becoming more and more common. You can study IT in 2-year technical schools, 2-year community colleges, and 4-year colleges and universities. Identify one degree in each of these three categories, and compare the required courses.
2. Go to your IT staff at your school or place of employment. Ask the staff how many of them have 4-year degrees in IT, 2-year degrees in IT, and 4-year degrees in a related field (e.g., computer science, information systems).
3. Which of the IT careers described in this section involve programming of some kind? Rank the careers from those that will have the most amount of programming to the least amount of programming.
4. Why should a webserver administrator and web developer be different careers? As you saw in the subsections, they use different skills and knowledge. For an idea of how complex a webserver can be, visit the Apache site and look at the complexity of the documentation (http://httpd.apache.org/docs/2.4/).

1.4 A CAREER IN IT

You are reading this text because you are interested in IT. What does a career in IT entail? Is it just a field where you get to use computers all day long? Is it similar to the activities you might have undertaken at home when you built a PC or installed new software? Is it simply a 9-to-5 job where you go in, help people solve their technology problems, and leave?

There are many aspects to an IT career that you might not have thought of. This section explores several dimensions of the IT career. We look at continuing education, the ethical responsibilities of an IT person, and some of the social implications of our modern IT world to be aware of.

Did You Know?

There are an estimated 2.8 million people working in IT in the United States alone. Of those, more than 900,000 were hired in 2017!

1.4.1 Continuing Education

We have already used the term *continuing education*. Continuing your education makes you a *lifelong learner*. What does this mean? Table 1.5 lists a number of ideas behind lifelong learning. These items are alphabetized rather than being listed in order of importance.

TABLE 1.5 Different Activities for Being a Lifelong Learner

Apply What You Have Learned	Ask Questions
Continue your education through classes, certifications, and online materials	Discuss ideas with others
Go to the library	Join groups to work on projects
Keep an open mind	Learn new technologies
Read (anything!)	Research topics online
Start your own projects	Teach others
Try new things out	Use your job as a learning environment

The phrase *lifelong learner* implies that you will continue to study throughout your life. This is critical for an IT professional because IT continues to change, perhaps more rapidly than any other field of study. In this author's adult lifetime, technology has seen PCs go from a hobbyist toy to a multibillion dollar industry. The microprocessor has progressed from running one program slowly to consisting of multiple cores that can parallel process and run graphically intensive tasks quickly. Storage technology has changed from floppy disks that store 1 MB to hard disks that can store more than 1,000,000 times that quantity. Mainframe computers have gone from the only source of computing to being largely obsolete, while computers in general have reduced in size from that of large rooms to that of a person's hand. The Internet has gone from a tool for academicians and researchers to a portal by which half the planet communicates, gets news, is entertained, and goes shopping.

I (your author) graduated with a bachelor's degree in computer science in 1986. If I had not expanded my own education, I would be as obsolete in the field as those computers of that era are now. Advances in IT have not been limited to just one area. As processor technology has progressed, so has that of applications and systems software. And to better support the creation of software, programming languages have progressed. Similarly, computer network technology and the Internet, as well as storage technology, has continued to change just as rapidly.

The IT specialist must learn, adapt to, and use new technologies. The IT professionals that do not continue to learn and adapt not only hold themselves back but also hold back their companies. Any employee holding back their company becomes a liability that could face an eventual layoff.

For IT professionals, continuing education is not merely a matter of remaining valuable to an organization but also a matter of helping the organization make informed decisions on future technology directions. There are numerous forms of continuing education available in IT. Most of these forms revolve around learning independently. There are many sources available to learn from, but it is up to you to take the initiative to find and use those sources. Among the forums available are trade magazines and journals, whether they are part of the popular press like *Wired*, *New Scientist*, *Dr. Dobb's Journal*, or more academically oriented such as IEEE and ACM publications. Table 1.6 provides a list of some of the more relevant IT publications from IEEE and ACM.

Two problems with reading the more academically focused journals are that they tend to emphasize cutting-edge research rather than the technology available on an industry level, and they are presented at a level that might be more amenable to graduate students and faculty. As an IT professional, you might prefer instead to research technology through the wide variety of books and websites. There is a large market in technology books. In fact, just about every IT topic will have several dozen to hundreds of books published on it, from users' guides to "dummy guides for" style books. Among the more popular IT publishers are Thompson Course Technologies, O'Reilly, Elsevier, and Springer Verlag.

TABLE 1.6 ACM- and IEEE IT-Oriented Journals

Journal Name	IT-Related Topics
ACM Transactions of Information and System Security	Cryptography, authorization mechanisms, auditing, protocols, threats, data integrity, privacy issues
ACM Transactions on Internet Technology	Digital media and digital rights, electronic commerce, Internet crime and law, Internet performance, peer-to-peer networks, security and privacy
ACM Transactions on Storage	Storage systems and architecture, GRID storage, storage area networks, virtualization, disaster recovery, caching
ACM Transactions on the Web	Browser interfaces, E-commerce, web services, XML, accessibility, performance, security and privacy
Communications of the ACM	Communications/networking, computer systems, computers and society, data storage and retrieval, performance, personal computing, security
IEEE Internet Computing	IT services, World Wide Web technologies
IEEE Networks	Network protocols and architectures, protocol design, communications software, network control and implementation
IEEE Technology and Society	Impacts on society, history of social aspects, professional responsibilities
IEEE Transactions on Computers	Operating systems, communication protocols, performance, security
IEEE Transactions on Dependable and Secure Computing	Evaluation of dependability and security of operating systems and networks, simulation techniques, security, performance
IEEE Transactions on Wireless Communications	Theoretical and practical topics in wireless communication, including prototype systems and new applications
IT Professional	Organizing data, cross-functional systems, IT breakthroughs, capitalizing on IT advances, emerging technologies, e-commerce, groupware, broadband networks, security tools

Note: For more information on IEEE, see www.ieee.org, and for more information on ACM, visit www.acm.org

Many technology-oriented websites are created and maintained by IT professionals. They publish their knowledge as a service to others who can benefit by learning from them. There are, for instance, a large number of websites that include information about the Linux operating system, the Windows 10 operating system, the Internet, the Apache web server, and so on. These sites can not only provide "how to" knowledge but also up-to-date tips of known problems and security patches.

If you are not comfortable learning independently, guided forms of continuing education are available. One form of continuing education is taking coursework. You can find many courses online and some for free. Massive open online courses (MOOCs) are being offered by a number of universities. Or, you can go back to school and take courses through a technical school or seek an advanced degree from a university. Your own organization might offer courses or hire a consultant to come in and teach you on a new technology. You can also take courses that specifically prepare you for certification. Many companies will support their IT professionals by paying for various forms of continuing education. For instance, the company might pay for your tuition in seeking either an advanced degree or a certificate. A variety of certifications are listed in Section 1.3. When you interview for IT jobs, you might ask what certifications the organization finds valuable.

There are also trade conferences taking place all over the world several times per year. These conferences include those sponsored by Adobe, Apache, CISCO, Dell, IBM, IEEE, Microsoft, Oracle, Red Hat, and VMware. Others cover topics such as cloud computing, Internet of Things (IoT), big data and machine learning, web services, databases, networking, virtualization, interoperability, emerging technologies, cryptography, and IT best practices to name but a few. As with continuing education courses, it is common for companies to pay for their IT professionals to attend such conferences.

In essence, to continue your IT education, you have to be willing to learn. Hopefully, you are already a self-learner and continuing your education, whether because your employer finds it valuable or because you are curious, will be something that you will enjoy.

TABLE 1.7 10 Commandments of Computer Ethics

1. Thou shalt not use a computer to harm other people.	6. Thou shalt not copy or use proprietary software for which you have not paid.
2. Thou shalt not interfere with other people's computer work.	7. Thou shalt not use other people's computer resources without authorization or proper compensation.
3. Thou shalt not snoop around in other people's computer files.	8. Thou shalt not appropriate other people's intellectual output.
4. Thou shalt not use a computer to steal.	9. Thou shalt think about the social consequences of the program you are writing or the system you are designing.
5. Thou shalt not use a computer to bear false witness.	10. Thou shalt always use computers in ways that insure consideration and respect for fellow humans.

1.4.2 Computer Ethics

Obviously, computers play an important role in our society. IT specialists must become aware of the impacts that computers, and IT in general, have on society. Issues include maintaining proper ethical conduct, understanding legal issues, enforcing IT-related policies, maintaining data integrity and security, and identifying risks to data and employees. Not only will you have to understand these for yourself, but you may also be responsible for the development, implementation, and education of company policies.

Table 1.7 lists the 10 Commandments of computer ethics, as published by the Computer Ethics Institute in Washington DC (from http://computerethicsinstitute.org/home.html). Although computer ethics is nothing new (the concept originated in 1950), with IT as a growing field, the notion of ethical conduct must be expanded to include those in charge of computer systems rather than those who create computer systems. IT specialists have greater privileges with respect to the computer systems because they have access to many or all resources. Therefore, they must act in an appropriate manner so that they do not violate the trust that the employers place in them to supervise these systems.

Aside from the 10 Commandments, other organizations have provided computer ethics codes of conduct. These include the Association for Computing Machinery (ACM), the British Computer Society, and the Institute of Electrical and Electronics Engineers (IEEE). These codes of conduct primarily relate to programmers and emphasize professional competence, understanding the laws, accepting professional review of work, providing a thorough risk assessment of computer systems, honoring contracts, striving to improve one's understanding of computing and its consequences, and so forth. These codes combine notions of ethics because they pertain to technology as well as ethics of being a professional.

It is up to the IT staff to implement the IT policies of an organization. IT personnel might help create such policies. Many policies revolve around securing the computer systems and the accumulated data that the organization maintains. We introduce a number of areas under which there should be policies in Table 1.8. For each, we provide brief explanations of the issues. We examine some of these topics in more detail in Chapter 9.

Aside from being charged with maintaining the privacy and protection of accumulated data, the organization must also act to preserve the privacy of the employees. Toward that end, IT will have policies regarding access to accumulated files. However, as an IT member, you may have access to files beyond your own. At what point do you examine the content of the files you have access to? IT policies may cover all of the content accumulated. For instance, do you have the rights to examine an employee's web browsing behavior or personal storage space?

Let's focus on ethical dilemmas that might arise for an IT professional. Afterward, we briefly examine some policy issues that an IT professional might be asked to provide in an organization. For the following situations, imagine yourself in the given position. What would you do?

TABLE 1.8 IT Policies

Area	Issues	Mechanisms
Privacy	• Ensure the privacy of data collected by the organization • Ensure employee privacy	• Assign levels to data • Proscribe access control rights based on employee role
Ownership	• Have a statement of client privacy with respect to accumulated data • Have clear-cut policies with respect to intellectual property as created by employees	• Disseminate privacy policy online or via postal mail
Control	• Delineate duties across IT staff	• Proscribe access control rights based on employee role
Accuracy	• Ensure accuracy of collected data • Have mechanisms whereby erroneous data can be corrected by clients	• Use automated and manual auditing for tracking changes to data • Have online or postal mail methods for clients to update erroneous records
Security	• Ensure secure storage for data • Ensure backups of data	• Use best practices with respect to network access • Use RAID and backup technologies

- Your organization uses a proxy server (this type of server caches web pages recently viewed by people in your organization). Your boss has asked you to explore the cached content to see if any of that content is pornographic in nature. Is it ethical for you to search through the proxy caches?
- Your boss suspects specific employees of illegally downloading content to their own file space. Your boss asks you to look at the content of these employees' directories. Is this ethical?
- You have noticed that the e-mail server is running out of disk space. You take it upon yourself to search all employee e-mail to see if anyone is using company e-mail for personal use. Is this ethical?
- You are a webserver administrator (not a web developer). One of the sites that your company is hosting contains content that you find personally offensive. What should you do? If the content is not obscene by the community's standards, would this alter your behavior? What if you feel the content would be deemed obscene by your community?
- You are a web developer. You have been asked by your employer to produce some web pages using content that they provide you. You believe the content is copyrighted. What should you do?
- You work for an organization that is readying a product for delivery. Although you are not part of the implementation team, you have heard from others that the product is not yet ready for release. Management is pressuring them to release the product anyway, bugs and all. What should you do?
- Your boss asks you to upgrade the operating system to a newer version that has known security holes. What should you do?
- You fear that your boss is going to fire you because of a number of absences (all of which you feel were justified). You decide, as a precaution, to set up a secret account (known as a "backdoor"). Is this ethical? Is this legal?
- Your organization freely allows people to update their social media sites from their work computers. But you are worried that some of your employees are putting up too much personal information about themselves. Is it your place to get involved?
- Your organization has a policy prohibiting people from accessing social media at work. You notice one of your colleagues has modified his Facebook page during work hours. Should you report this?

As can be seen from these examples, you may be asked to put yourself into a position that violates employees' privacy or confidence. Or, you may be placed in a situation where your own morals conflict with the tasks assigned to you by your employer.

IT needs to have clearly defined policies for IT use and IT staff. Here are some ideas of IT policies.

- The organization allows or does not allow work computer usage for personal use. If personal use is allowed, reading personal e-mails, accessing social media, and visiting websites that have little or nothing to do with the organization may be permitted. If personal use is not allowed, something as simple as reading a personal e-mail would be a violation of this policy.
- The organization allows or does not allow personal mobile device use during work hours. If the organization does not allow you to use work computers for personal use, can you still access your personal e-mail and social media via your own mobile device during work hours?
- IT personnel are assigned a level of clearance and may view all content within that level. Alternatively, you may be assigned a level of access but not permitted to view specific content unless authorized.
- The organization requires or encourages whistleblowing, possibly anonymously. As an IT person, you may see things that others do not see about employee IT use. Are you required to bring this to management's attention? If so, are there anonymous ways to do so?

  When asked to monitor employees, you are being asked to spy on them. This would be considered ethical if IT policy clearly states that the organization has the right to inspect employee IT use and enumerates activities that employees are not allowed to do with the organization's IT. In such a situation, it would be ethical to do so.

If your employer tasks you with monitoring employees, and you feel that such an activity is not defined or expected based on IT policies, your ethical stance becomes less straightforward. The best response is to ask your employer to update the IT policies so that the employees understand the parameters that they are asked to work within. On the other hand, even if the organization has well-defined IT policies, they need to be clearly disseminated to the employees. If employees are unaware of the policies, the policies do little good.

Some of the IT-related questions that should been stated formally as policies follow. As an IT person, you may be asked to write or help write such policies.

- What is the role that IT plays in the company?
- What is considered proper employee use of IT?
- What are the employee's rights to privacy (if any)?
- What is the proper use of accumulated data?
- How will accumulated data be kept secure?
- How will stored data be maintained?
- What are the ownership policies for employee-created intellectual property?
- How does the company handle copyright protection issues?

Drafting policies may be a foreign concept to an IT specialist. But at a minimum, IT personnel should help management draft the policies because management may not be aware of technology best practices available to implement policies. Such technology might include the frequency and storage of backups, the use of RAID technology, security concerns with webservers and file servers, network intrusion detection systems, the use of data mining on collected data, and the use of tracking software to monitor employee computer use, just to name a few.

Just as an IT specialist should understand both the policies of the organization as well as the proper ethical stance, understanding the laws is equally important. Consider a situation in which your boss has asked you to install software that he hands to you on a flash drive. Was this software purchased legally? Are there licenses with it? If not, do you go ahead with the installation? Just because your employer has asked you to perform a task does not mean that the task is either ethical or legal. Knowing the law can not only protect you but can also help you inform your employer about potentially illegal situations. Unfortunately, understanding laws that regulate IT and software can be challenging. Even if you can identify the laws, they are often written vaguely and in "legalese".

Consider as an example that your company has created a policy whereby all employees will be known by their social security number. That is, the social security number will serve as their unique identifier for database entries. As such, it is also used in their log in process. Because it is part of their log in, social security numbers are stored in one or more files in the operating system that might be accessible to the IT staff, if not all employees. As the system administrator, what will you do? Is the use of social security numbers a violation of a company policy? No. Is it illegal? Not strictly speaking. Is it unethical? Probably.

You convince management that using the social security number for log in purposes is a bad idea, and therefore, this secure data is removed from login files. Yet, the social security numbers are still stored in company databases because of payroll and tax purposes. Database managers and the payroll staff have access to this data. Is this use of the social security number illegal? No, actually the social security number is required by law. Is this unethical? No, because those who have access to the social security numbers are those that the company deems as having a legitimate need to access such data. Just as long as the company has ensured that the storing of social security numbers is secure, the company's use of the social security numbers is ethical.

The IT staffs have the added responsibility of protecting devices that store all sensitive data. This not only includes file servers or database servers but also mobile devices. Consider that employees are taking a business trip. They load secure data files onto a laptop to work on while away. The laptop is lost or stolen during the trip. This puts the company in a bad situation. How can this situation be prevented? Two possible solutions are to encrypt the data so that even if the laptop is stolen, the thief would be unable to access the data or to keep the data within the organization's network and have the employee remotely access the data through some secure login mechanism. Both solutions are not as efficient as having the data readily accessible on the laptop, but both ensure the security of the data.

1.4.3 Other Social Considerations

In this subsection, we go beyond legal and ethical considerations and organizational policies and consider some key social concerns regarding IT. Although IT personnel may not need to bring these to the organization, it is useful to be aware of some of the many technology-related problems in our world today.

We start with the *digital divide*. There are gaps in access to computers and the Internet based on people's economic, demographic, geographic, and educational backgrounds. Gaps between technologically advanced and wealthy nations over poor nations continue to grow. Because the Internet has afforded those who have access a tremendous amount of information at their fingertips, those without or with limited access become even more disadvantaged with respect to an informed life, a healthy lifestyle, education, and employment. Governments and multinational corporations have addressed the digital divide, but it continues to be a concern. Although, as an IT specialist, you may never have to deal with the ramifications of the digital divide, understanding it might help your organization create reasonable policies.

Table 1.9 compares the number of Internet users by percentage of population for countries of sizeable populations. The listing is from the greatest percentage of population access to the least. The table also provides a relative economic ranking (gross domestic product) of high, medium, or

TABLE 1.9 Internet Use for Larger Countries

Country	Access (%)	Relative Economic Level
South Korea	95.1	Medium
United Kingdom	94.6	Medium
Canada	92.7	Medium
Japan	90.9	Medium
Spain	84.6	Medium
Germany	84.4	Medium
France	80.5	Medium
Russia	76.0	Medium
United States	75.2	High
Brazil	67.5	Medium
Turkey	64.7	Low
Mexico	63.9	Medium
Italy	61.3	Medium
Iran	60.4	Low
Philippines	60.0	Low
China	54.3	High
Thailand	52.9	Medium
Vietnam	49.6	Low
Egypt	45.0	Low
India	34.5	Medium
Indonesia	32.3	Medium
Nigeria	27.7	Medium
Pakistan	15.5	Low

low. Numbers are estimates from 2017. Notice that the countries listed with the lowest percentage of Internet access have medium or low economic rankings.

Another issue that any IT person should understand is electronic waste, or e-waste. *E-waste* is the careless discarding of electronics. We tend to throw out old electronics equipment, and it often ends in a landfill. The problem with this is that electronics components include a wide variety of chemical elements that are toxic to the environment. Mercury, for instance, might leak into the soil and then into groundwater, poisoning the water and any fish that might live in that water table. Table 1.10 lists some of the more toxic chemicals, along with nontoxic chemicals, found in electronics equipment.

We produce a substantial quantity of e-waste. It is estimated that the world produces between 20 and 50 million tons of e-waste a year. This is largely caused by how quickly electronics devices become obsolete. On average, a person purchases a new smartphone every couple of years and new laptops and desktops every three to five years. We throw out old television sets, stereos, speakers, and microwave ovens.

What can we do about e-waste? The most important and responsible action is to recycle electronics. There are companies that will take electronics when a replacement is bought. In some cases, a discount is sometimes given on a new purchase in this arrangement. These companies may donate the electronics to organizations that could use them. For instance, a three-year old computer may still be of use to a grade school. Alternatively, the companies may disassemble the traded-in electronics, discarding the harmless components, and properly disposing the components that have toxic elements. The harmful components may wind up in an e-waste lab.

On the other hand, it has become common practice to collect discarded electronics and ship them to third-world countries. These countries take the electronics because they are paid to do so. And then they wind up in landfills, or worse, they just get piled up. Figure 1.4 shows a couple of disturbing pictures of mountains of electronics.

TABLE 1.10 Potentially Toxic Chemicals in Electronics

Chemical Element	Used In/As	Is It Toxic?
Barium	Magnetic devices (e.g., magnets, magnetic strips on credit cards), spark plugs	Partially because it can impact your ability to breathe, impact blood pressure, and cause swelling in key organs
Cadmium	Batteries, cathode ray tubes, solder	Yes, in large enough quantities it can poison kidneys and liver and bring about lung damage or cancer
Hexavalent Chromium	Coating of metals to prevent corrosion; also used in paint	Yes, it can cause many forms of cancer as well as ulcers and problems with the intestines
Lead	Circuit boards and chips, wire	Very, particularly to children and pregnant women where lead can cause brain damage of the unborn fetus; other impacts are increased blood pressure and kidney damage
Mercury	Batteries, circuit boards, manufacturing of electronics	Very, it can cause brain damage and kidney damage
Nickel	Batteries, wire, coating of steel	Some nickel compounds are carcinogenic
Polyvinyl chloride	Plastics that coat cables and computer components (e.g., monitors)	Increases the risk of several forms of cancer, particularly harmful to children
Zinc	Batteries, displays, combined with other elements like nickel	In large enough quantity, it might lead to cancer

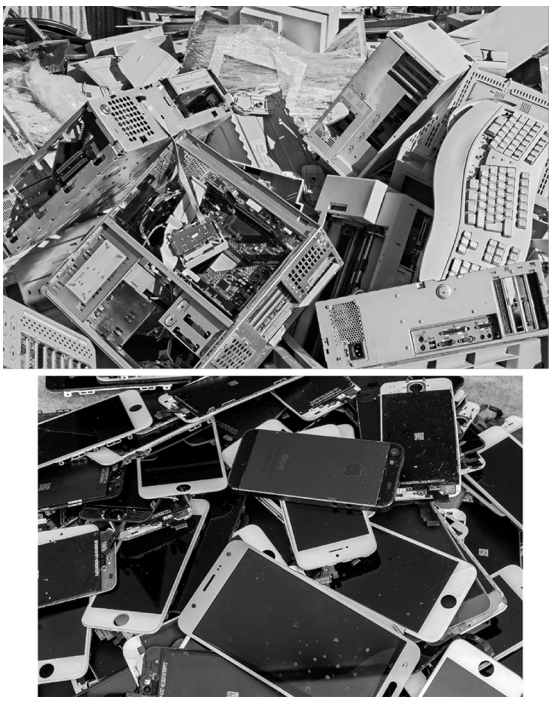

FIGURE 1.4 E-waste discarded into landfills.

In truth, although people are trying to be responsible, it is estimated that less than 20% of all electronics are collected for recycling, and as much as 50% of all electronics that are collected are improperly discarded. There are a couple of solutions. First, governments should give companies incentives for the proper disposal of electronics so that those companies can provide incentives for users. Second, we as a society should not be so quick to replace our electronics. Additionally,

manufacturers are exploring alternative chemicals to produce electronics, but it is a challenge to find a suitable, nontoxic replacement, particularly one that might be cost effective.

The next issue is perhaps a problem not so much because people have access but because of what they do with that access. In short, social media has given every person who has access to the Internet a means of providing information about themselves, whether it is pictures, biographical information, opinions, or some combination. The problem we have today is that the amount of information we put out into publicly accessible forums can tell a lot about us. This has been termed *digital litter*.

Consider a person who tweets to his friends that he and his family are leaving town for a week-long trip. It doesn't take much effort to identify the person's address, and knowing that the house will be empty for some time, a thief might decide to burglarize that person's home.

Understanding the threats of such self-advertising is important to all computer users in this age of *identity theft*. We might be giving out far more information than we expect, even if we think we are only sharing such data with a few select friends. There are websites that accumulate masses of information on everyone. You can view some of this information for free. But if you subscribe to one of these sites, you can find a great wealth of information. Table 1.11 provides a listing of some of the information that might be gathered about you and the sources of that information. Note that a source of "social media site" means that you put the information out there yourself.

It's frightening how much data might exist about you because a large set of data is deemed publicly available. This is true of most data that the federal government might compile on a person. But it doesn't help the situation if you freely advertise more information about yourself by posting on social media. Think about what you might have already posted or said on social media. Have you talked about politics or religion? Your teachers or employers? Trips that you have taken? Hobbies? Health issues or concerns? Your economic status? Accidents you may have been in or caused?

Let's also not forget that what is posted on the Internet may stay there a long time. Even if you upload content and later delete it, copies exist. Tweets can be retweeted. Pictures can be copied and posted on other sites. Social media sites regularly perform backups so even once you delete something; it exists in a backup form somewhere.

If you haven't already done so, explore The Wayback Machine at archive.org/web. Entering a URL into their search box brings up a listing of instances of when that website was stored by archive.org. You can, for instance, view a website as it looked 15 years ago. You can find postings on social media sites or stories from news sites that have long since been taken down.

TABLE 1.11 Types of Digital Litter

Datum	Source	Privacy Level
Full name	Just about anywhere	Public
Date of birth	Birth certificate	Public
Address	Yellow pages, mortgage document	Public
Property ownership, property value	Mortgage document, property tax data, real estate websites	Public
Phone number(s)	Yellow pages	Public (unless unlisted)
Relationship status	Marriage license, social media site	Public, available via social media
Employment	Employer website, social media site, public registry if you work for the city, state, government	Varies by your employer but could be public
Religion	Parishioner registry, social media site	Should be private but you might make it public
Political affiliation	Social media site	Should be private but you might make it public
Photos	Social media site	Should be private but you might make it public
Children, siblings	Yellow pages, birth certificate registry	Public
Hobbies	Social media site	Should be private but you might make it public

Threats can go far beyond losing your privacy. We mentioned identity theft, whereby someone learns enough about you to convince others that they are you. The most common use of identity theft is to obtain a credit card in your name and use it to purchase items at your expense. Phishing attempts (discussed in Chapter 9) are also prevalent. Scammers attempt to obtain secure information about you to be used against you. A common approach is a phone call or e-mail claiming to be from an authorized IT person telling you that IT has lost pertinent data (such as your password or social security number) and that you must provide the caller with this information or else your account will be disabled. Knowing that you use a particular website tells the scammer that you might be susceptible to a phishing attack.

Maintaining our privacy is critical in this era. Digital litter is opening up your privacy to others. Even if you trust your Facebook friends, what if they have friends that you do not know, and they share some pertinent information with one of them?

Many of us are addicted to our smartphones. How often do you look at it to see if an e-mail or text has arrived? How often do you visit your social media site(s)? *Internet addiction* is the impulse or need to be online frequently. Health organizations have even created a term for this: *Internet addiction disorder.* The number of hours that many people are now spending using computers has surpassed the number of hours that they view television. Much of the computer use revolves around Internet access.

You have probably noticed this when you are out in public or at home with the family (you might even be guilty of this yourself). In social events like the movies or a concert or sporting event, how many people pull out their phones? How often do you pull out your phone at home while watching television or eating? If you are not alone, this might be considered rude behavior!

There are several forms of Internet addiction. The first is net compulsion. Here, the person has the need to check on the latest status of something whether that is sports scores, stock prices, or recently received e-mail. Net compulsion could potentially lead to online gambling. The second is the need to communicate to online friends. This might be in the form of social media posts, e-mails, or texts. The danger here is that you might be spending more time with online communication than you are with in-person communication. A third form is in information seeking. When a question comes up, do you grab your phone to search for an answer? We are being driven by an urge to learn trivia at an increasing and alarming rate. Fourth, and perhaps worst, are those who have become addicted to activities online in the form of gaming or pornography. Both are considered harmful because they remove you from real-world interactions.

Mentioning pornography leads us to another concern: improper use of our ability to access content on the Internet. Specifically, we need to understand *obscenity* and *indecency.* The Internet has opened up the ability for people to read and view almost anything on the planet. Content may include hate speech and militant ideologies, pornography, violent video content, and the knowledge to build a bomb. The problem that we have is much of this information is not available within our local community because our community might deem such information indecent or obscene. Indecency and obscenity are defined based on your own community's sense of morality. Thus, what is considered indecent to people in rural America may differ from those in Los Angeles. The Internet, being global, gives people access to material that their local community might condemn but others find unobjectionable.

Obscenity has long been an issue in our society. Rating services, the Federal Communications Commission (FCC), and watchdog organizations attempt to police movies, television, and radio. However, policing the Internet is an entirely different matter because of its global nature and enormous content. Censorship might be implemented to prevent material deemed obscene from being viewed. However, the measures to censor websites are crude. For instance, you can block all content from a particular website or block content based on specific keywords. But how do you block a picture of a naked person unless you block all pictures? It is not up to your Internet service provider to block content. You might attempt to block content within your household, just as public libraries might block some content.

The Chinese government has taken the matter to an extreme level by blocking certain content coming from the Internet into their country. That is, they have created a firewall around all of Chinese Internet access. This is known as the *Golden Shield Project*, but the more popular term is the *Great Firewall of China*.

Our last concern has nothing to do with the Internet or data but is in fact health related. Extensive computer use can cause any number of health problems. The most common issue is the repetitive stress injury (RSI) that arises from performing the same physical motion over and over again. One of the most common forms of RSI is carpal tunnel syndrome, a muscular problem that impacts the wrist. This is often caused by extensive misuse of the keyboard and mouse.

Aside from RSI, back strain, sore shoulders and elbows, eye strain, and even hip and knee strain are often attributed to heavy computer use. Other problems with heavy computer use include headaches and stress. These problems are well documented, and there are several solutions. *Ergonomics* and *human-computer interaction* (HCI) offer us solutions. Ergonomics is the study of designing equipment and devices that better fit the human body. You might, for instance, have a wrist pad for your keyboard or have a special chair that supports your back better. HCI informs us of better colors to use to reduce eye strain and headaches and to group content together so that the user has less to look at or less distance to move the mouse.

Despite ergonomics, people continue to have health problems because of poor posture or just simply because they are not remembering to take breaks from their computer. We would all be better served to stand up and stretch and take a five-minute break from our computer every hour or more often. We return to HCI briefly in Chapter 3 when we look at input and output devices.

SECTION 1.4 ACTIVITY

1. Go to monster.com or indeed.com or some other employment website and type in "information technology" and your city. How many IT jobs were found? Use system administrator instead, and then try database administrator followed by software developer. Which job has the most listings?
2. Have you taken a course in ethics? If so, do you feel that IT should have its own area of ethics or is ethics, as applied to a person, suitable for people specifically in the IT field?
3. Which side of the digital divide are you on? Does your household have several computers and smartphones? Do you have broadband Internet access? How many people do you know on the other side of the digital divide?
4. Have you discarded an electronics device in the trash? Has your family? Or have you taken an old device and given it to an electronics store or recycling company to recycle it?

See the textbook's companion website for the chapter review material: www.crcpress.com/9780367820213.

2 The CPU and Memory

In this chapter, we begin to explore computer systems in detail. We examine the processing portion of the input, processing, output, and storage (IPOS) cycle. Processing involves both the processor and memory. The processor or central processing unit (CPU) executes (processes) program code. For efficiency, we store our program instructions and the data that the program operates on in memory. The CPU accesses memory continuously to execute the program. Unfortunately, our computers' main memories, although large, tend to be slower than the processor and so to keep the processor running smoothly we provide computers with multiple memories of different sizes and speeds. These memories are collectively called the *memory hierarchy*. In this chapter, we explore both the CPU and the memory hierarchy and also introduce the pipeline to improve processor performance and ways to evaluate a computer's performance. We conclude with an examination of the evolution of computer technology. The learning objectives of this chapter are to be able to:

- Explain the role of the central processing unit (CPU)
- Describe the components of the CPU in terms of their function, how they work, and what they communicate with
- Explain the use of the various buses
- Explain the types and roles of the various types of memory in the memory hierarchy
- Determine the efficiency of a computer based on the type and speed of CPU and type, size, and speed of the various components of the memory hierarchy
- Describe the evolution of computer technology and the impact of each type of technology

2.1 AN OVERVIEW OF A COMPUTER SYSTEM

Figure 2.1 illustrates the components of a typical computer system. We see three boxes that represent the main components of our computer: the CPU, memory, and input/output (I/O) system. Connecting these components together is a bus called the *system bus*. Each of these three components and the bus are all composed of lesser components. To understand the computer, we need to look at the CPU, memory, I/O, bus, and their component parts. In this section, we provide a brief overview defining the role of these four components. In Sections 2.2 and 2.3, we explore the CPU in detail. We also look briefly at the bus in Section 2.2. In Section 2.4, we look at the various forms of computer memory and how these memories interact with each other and the CPU. Section 2.5 describes how we can make our computers more efficient along with ways of gauging efficiency. Section 2.6 concludes with a history of computer technology. We hold off discussing I/O devices until Chapter 3.

The way the computer works is as follows. A program is loaded into memory. The program has already been compiled into an *executable* form (i.e., the program is written in the processor's native machine language). The executable form contains both program instructions and space reserved for data that the program instructions use. Thus, both the program's code and data are stored in memory.

Memory is broken into distinct locations, each with its own *address*. The CPU is informed of the address of the program's first instruction. The CPU begins executing the program using what we

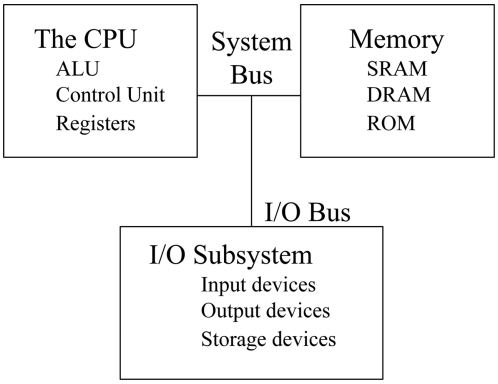

FIGURE 2.1 Basic view of a computer's components. ALU, arithmetic-logic unit; CPU, central processing unit; DRAM, dynamic random access memory; I/O, input/output; ROM, read only memory; SRAM, static random access memory.

call the *fetch-execute cycle*. One cycle consists of fetching one program instruction from memory, bringing it into the CPU where the CPU can determine what the instruction requires, and executing the instruction. Note that a fetch-execute cycle is not the same as a clock cycle. We explore the difference in Section 2.5.

The fetch-execute cycle's first step is to *fetch* the next program instruction from memory. As the program starts, this will be the first instruction, whose address is known to the CPU. The fetch step requires performing a *memory read* operation. Memory reads are handled by the CPU sending the address of the item it wants to memory along with a *read command*. Upon receiving the read command and an address, memory accesses that address, retrieves the contents, and sends it back to the CPU. The CPU receives the contents, an instruction, and stores it internally. The CPU then increments the address for the next instruction so that it is ready to fetch the next instruction on starting the next fetch-execute cycle. The idea of fetching an instruction and incrementing the address is shown in Figure 2.2. Notice that the instruction itself is a string of binary digits. We will refer to this in the following section.

With the fetch complete, the CPU examines the fetched instruction to determine precisely what the CPU needs to do to execute the instruction. This step is known as *decoding*. This instruction is written in the processor's machine language. The instruction comprises multiple parts. At a minimum, the instruction contains the *op code*, which is the binary encoding of the operation to execute. There may be further information encoded in the instruction such as the data that the instruction will operate on. We refer to the data as *operands*.

Different processors can accommodate a different number of operands. Some execute on a single operand, some on two operands, and some on three. Figure 2.3 illustrates the difference where we want to add two numbers together, X and Y, storing the result in Z. Notice that the three-operand format requires an extra instruction but is usually preferred for a number of reasons that we will briefly discuss in Section 2.5.

The operands can be stored in memory, presented as literal values, or stored in registers (storage locations inside the CPU, such as R1, R2, and R3 as referenced in Figure 2.3). The instruction needs to encode the information of where the operands are stored. We refer to the encoding of the location of data as the *addressing mode*. If the operand is stored as a literal value, then we use the *immediate* addressing mode. If the datum is stored in a register, we use the *register* addressing mode. If stored in memory, we supply the address of the datum (*direct* addressing mode) or the address of a pointer to the datum (*indirect* addressing mode). There are other modes that might be used for array values, strings, computer graphics, and other types of data.

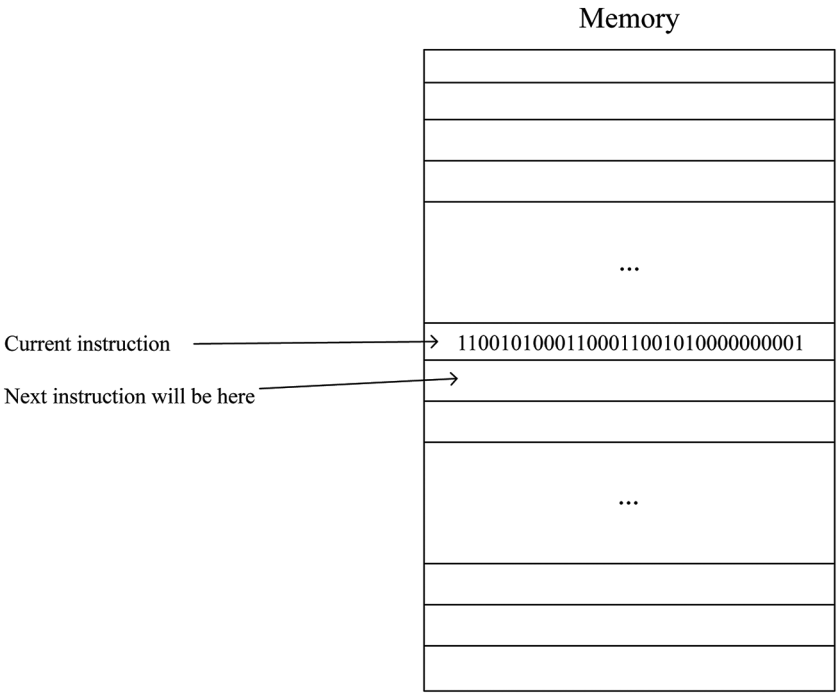

FIGURE 2.2 Fetching an instruction from memory.

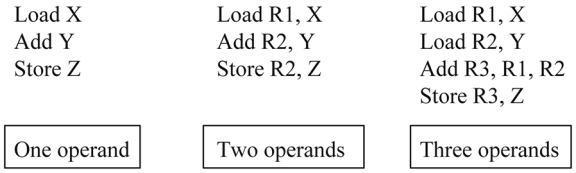

FIGURE 2.3 Referencing data in instructions.

The decoding step involves interpreting the bits of the instruction to identify the op code, and given the operation, the number of and location of the operands. From Figure 2.2, we fetched the instruction `1100101000110011000101010000000001`. Let's assume the op code is the first seven bits (`1100101`). Let's further assume that this op code represents the ADD instruction. Addition requires two data to add together (which we can call the *source* data) and a location to store the result (called the *destination*).

Next, the processor must determine where the source and destination are. In the case of a one operand instruction (see the left side of Figure 2.3), there is a single source with the other source and destination being a special register called the *accumulator* (which we often abbreviate as AC). This makes the instruction `AC = AC + datum` where datum is indicated by the one operand specified in the instruction.

If the instruction has two operands, then one is known as the source and the other is both a source and the destination. The instruction then is `destination = destination + source`. In both one- and two-operand versions of our instruction, we have a *destructive* operation because one of the original data is overwritten (the AC in the first case and the destination in the second). A three-operand version requires three different locations specified and is the only form that is non-destructive.

Returning to our example instruction, the next five bits encode the addressing modes used. The mode is `00011`, which we will assume means that the instruction uses three operands where the destination is a register, one source is a register, and the other source is an immediate datum. The next sequence of bits encode the destination register, `00110`, or register 6 (R6), and the first source, `00101`, or register 5 (R5). The remaining bits are the immediate datum, or a specific value to use. In this case, the value is `0000000001`, which is binary for the number 1.

Decoding results in the CPU dividing the instruction into the following parts.

`1100101` – ADD operation
`00011` – Addressing modes (3 operands, 2 registers, and an immediate datum)
`00110` – Destination register number
`00101` – Source register number
`0000000001` – Immediate datum (the literal number 1)

This instruction might be written in a more English-like way as `ADD R6, R5, #1`, meaning add the value in R5 and the value 1 and store the result in R6. In a high-level language, the instruction might look like `x = x + 1;` assuming *x* is currently stored in R5 and the resulting sum will be stored in R6.

Having decoded the parts of the instruction and knowing where the data are to be found, the next step is for the CPU to retrieve the data. In this case, one datum is in register R5 and the other datum is literally the value 1. If the instruction called for a datum to come from memory, then a memory access would be required. Register access is faster.

Once the CPU has the value from R5 and the number 1, it can proceed to *execute* the instruction. Execution depends on the type of instruction. An addition is handled by the circuitry in the arithmetic-logic unit (ALU). The CPU passes the two operands (the value stored in R5 and the number 1) to the ALU along with a signal to perform an addition. The ALU is a large circuit that operates on electrical current. Current flows into the ALU representing the two data and the result comes out of the ALU representing the result. In this case, the result is the sum of R5 and 1. The CPU moves the result to the destination register, i.e., register R6.

WHO MAKES CPUs?

First, we need to differentiate between those that design CPUs and those that manufacture them. AMD, for instance, designs CPUs and then sends them out to be manufactured. The two main companies that develop CPUs are Intel and AMD. Intel has the iCore processor but has also Itanium and Xeon (among others). AMD has Athlon, Phenom, and Opteron (among others). There are other companies that manufacture CPUs, including ARM Holdings (who design and manufacture ARM processors for mobile devices), Nvidia (who makes GPUs), Texas Instruments (who primarily develop processors for calculators and micro-controllers), and Qualcomm (who develop processors for telecommunications devices). But when it comes to our personal computers, it's pretty much Intel and AMD processors that we use.

With this fetch-execute cycle completed, the CPU repeats this cycle on the next instruction. Recall that the CPU had already incremented the address of the instruction so the next fetch can start immediately by fetching the instruction at this new address. The CPU performs the fetch-execute cycle on this new instruction. In doing so, it increments its address to point at the next instruction.

The steps for the fetch-execute cycle are as follows.

- Fetch the instruction from memory and increment the address of the next instruction.
- Decode the fetched instruction.
- Fetch operand values.
- Execute the instruction.
- Store the result (if any).

Every fetch-execute cycle requires accessing memory to obtain the instruction to be executed. In this example, the add operation used data from one register and one immediate datum, which is stored as part of the instruction itself, which we saw as the binary value `0000000001`. The result was stored in another register. Registers are part of the CPU. Some instructions might reference memory to obtain or store data rather than using registers. So, it is possible that a single fetch-execute might involve multiple memory accesses.

Because of this, the CPU and memory are linked together quite closely. Connecting them is the *bus*. The bus is an *information mover*. The bus consists of connections between components. In older computers, wires physically linked components together. Today, we use lines (possibly copper) etched on chips or the motherboard, one line per bit. In Figure 2.1, we saw that the CPU and memory are linked together by a bus.

Figure 2.4 shows a typical personal computer motherboard. The CPU is inserted into a socket (toward the right on the motherboard in this figure). Memory chips are placed on memory circuit boards and inserted into slots that run horizontally below the CPU socket. Connecting them together is the system bus, one of several buses in the computer. This bus is etched onto the motherboard (not visible from the top side of the motherboard). Other slots, which run vertically toward the left of the motherboard, are used to connect I/O and storage devices and it also connects to the bus.

Personal computer motherboards divide the system bus into two separate buses that go by the names *northbridge* and *southbridge*. The northbridge connects CPU and memory, and the southbridge connects I/O devices. More generically, these two buses go by the names *system bus* and *I/O bus*.

With this brief introduction made, we take a closer look at the CPU followed by memory in the next two sections. In Section 2.5, we also add the system clock into play and look at ways to determine the effectiveness of our CPU.

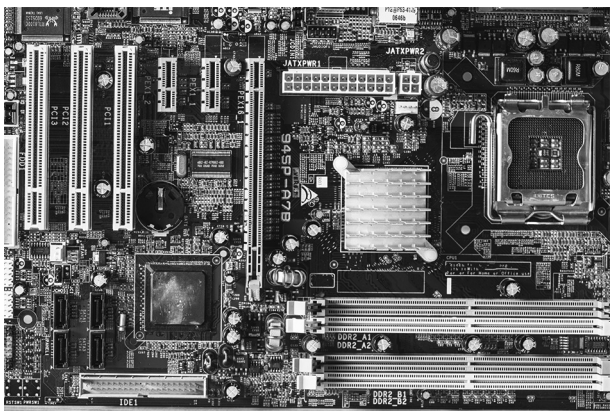

FIGURE 2.4 A motherboard.

SECTION 2.1 ACTIVITY

1. Determine the size of your computer's main memory. What does GB mean? Assuming we can store 1 character (e.g., letter, digit, punctuation mark) in 1 byte and that the typical book consists of about 200 words per page and the typical word is 5 letters, separated by spaces and punctuation, approximately how many pages could your computer's memory store?
2. The example instruction provided in Figure 2.2 is 32 bits long. Many processors use 32-bit instructions. Why do you suppose this size was chosen for instruction lengths? Hint: Think about the term "32-bit computer."
3. In Figure 2.4, try to identify the CPU's socket, the slots for memory, and the slots for I/O devices.

2.2 THE CPU

The CPU comprises at least three components: the control unit, the arithmetic-logic unit (ALU), and registers. See Figure 2.5 for a look at some of the details of the CPU. We will refer back to the specifics of this figure throughout this section.

2.2.1 Registers

Registers are used for short-term storage in the CPU. They are divided into two types: *control registers* used by the control unit and *data registers* used by the ALU. Data registers may be broken into two types as well which include those that store integer values and those that store floating-point (real) values. In some processors, there is a division in data registers between those storing data and those storing addresses. An address is used to indicate the location of a datum in memory. We use addresses when we are dealing with arrays, objects, parameters, and pointer variable types.

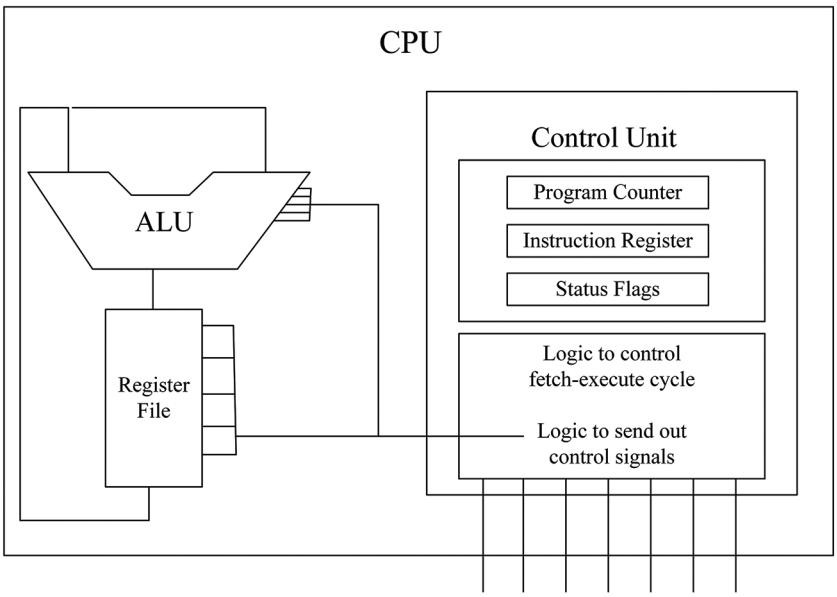

FIGURE 2.5 The central processing unit (CPU). ALU, arithmetic-logic unit.

The CPUs vary greatly when it comes to the number of data registers. Many early processors (as used in mainframe computers of the 1940s and 1950s and early personal computers) had a single data register named the *accumulator*. This register would be used in any and all arithmetic operations. The other datum for the operation might come from memory or be a literal value. We saw an example of this with the one-operand instructions from Figure 2.3. Let's revisit that set of code. We want to execute Z = X + Y. This is broken into three instructions.

```
Load X
Add Y
Store Z
```

Each of these instructions refers to one operand, which here is a variable in memory (X, Y, and Z). Each of these instructions needs at least one more operand. For `Load X`, we need a place to store the value of X in the CPU. For `Add Y`, we need a value to add to Y and a place to store the sum. For `Store Z`, we need an operand to move to memory location Z. In each of these cases, that extra operand is implied to be the accumulator. In effect, these instructions could be rewritten as follows.

```
Load AC, X
Add AC, AC, Y
Store AC, Z
```

We refer to a CPU with only the AC as a data register as an *accumulator-based processor*. The accumulator-based processor can be challenging to program because we will need to continually move data from memory to processor and back to memory as we compute other values.

Did You Know?

Many high-performance computers contain vector registers which are registers that can store several to dozens of array elements at a time.

Let's consider a slightly more elaborate example of Z = A * B + 3 * C. Because of operator precedence, we compute A * B and 3 * C before we can perform the addition. With only the AC, we have to compute A * B, move it back to memory to some temporary storage location, compute 3 * C, and then fetch A * B from the temporary storage location so that we can compute the sum. The code that can accomplish this is shown as follows along with the comments.

`Load A`	AC gets A
`Mult B`	AC gets A * B
`Store Temp`	Need to free up the AC, move A * B to Temp
`Load C`	AC gets C
`Mult 3`	AC gets 3 * C
`Add Temp`	We can perform the addition, AC = A * B + 3 * C
`Store Z`	Store the result

The inclusion of another variable in memory, `Temp`, is needed here to complete the operation. A processor with more data registers improves both our ability to write code for the processor and

the processor's efficiency. We rewrite this code as follows, assuming we have a *register file*. This is a set of registers to store multiple values. We refer to each register in the register file by a different name, for instance R0, R1, R2, etc. This code contains one fewer instruction and two fewer references to memory so it can be executed more efficiently.

```
Load R1, A          R1 gets A
Mult R2, R1, B      R2 = A * B
Load R3, C          R3 gets C
Mult R4, R3, 3      R4 = 3 * C
Add R5, R2, R4      R5 = A * B + 3 * C
Store R5, Z         Store the result
```

Reexamining Figure 2.5, you can see that the register file has a pathway to the ALU so that values stored in registers can go to the ALU, and the ALU has a pathway to the register file so that results can be stored in registers. Coming from the control unit is a series of lines that tell the register file which register is to be used. In the figure, there are five lines going into the register file. This allows for us to select between 32 registers ($2^5 = 32$). It is common for CPUs to have at least 32 data registers.

2.2.2 The ALU

The ALU is a collection of circuits that each performs a different arithmetic or logic operation. The arithmetic operations are, at a minimum, addition and negation. Addition is self-explanatory. Negation takes a number and creates its negation, for instance, converting +5 to −5 or −5 to +5. With these two circuits, we can implement subtraction. The process for X − Y becomes X + (−Y). Most likely, our computer has a single adder/subtractor circuit that can perform either addition or subtraction because the adder/subtractor has a built-in negation circuit.

Old processors performed multiplication as a sequence of additions. For instance, 3 * 4 would be calculated as 3 + 3 + 3 + 3. Old processors similarly performed division as a sequence of subtractions. The quotient is the number of times the denominator can be subtracted from the numerator until the numerator is less than the denominator, and the remainder is the value left over after the last subtraction. For instance, 14/3 requires that we subtract 3 from 14 repeatedly. We can do so 4 times, leaving 2, which is the remainder. Today, multiplication and division circuits are part of the ALU where the division circuit computes both the quotient and the remainder at the same time.

We divide the arithmetic operations between integer-based and floating-point-based. So, we actually need twice the number of circuits described previously, one set for integers and one set for floating-point. Just as early processors did not have multipliers and dividers, most early processors, particularly those of personal computers, did not have floating-point circuitry. Instead, floating-point operations were computed using a combination of the integer-based components and software.

If you preferred faster floating-point computations, the alternative was to purchase a math coprocessor such as the Intel 8087 or the Intel 80387. Most early personal computers had an extra CPU slot for the coprocessor, which would be used only when floating-point operations were needed. Starting with the Intel 80486-DX chip, math coprocessor circuitry was added to the standard CPU. Starting with the Pentium processor, floating-point circuitry became standard. They referred to these chips as Pentium-MMX, where MMX stands for multimedia. Multimedia primarily meant computer graphics, which consists of many floating-point operations.

INTEL PENTIUM BUG

Intel released their first Pentium processor in 1993. Unfortunately, it contained a bug in its floating-point division circuitry. The bug was discovered by a professor at Lynchburg College in Virginia in 1994, and Intel released a fixed version later in 1994. To reduce the time for floating-point division, a lookup table is used to retrieve certain quotients that are used during the division operation. The lookup table was computed in advanced but in transcribing it into a ROM, some of the values were omitted. This resulted in errors in certain floating-point divisions but not all floating-point divisions! Intel later reported an approximate $475 million loss in replacing all of the processors.

Aside from arithmetic operations, ALUs also perform bit-level operations. We divide these into two categories: shifts/rotates and logical operations. Shifting and rotating slide bits in one direction or the other. They are similar, except that rotating bits causes the last bit to wrap around to the other end. We cover shifting (although not rotating) in Chapter 4.

The logic operations are AND, OR, and NOT. From these three basic operations, we can also build other logic operations called XOR, NAND, NOR, and COINCIDENCE. We explore the role of some of these in Chapter 4 and omit any details here.

One additional logic operation is *comparison*. The comparator circuit is used to determine the relationship between two numbers. If we have X and Y, we might want to know if X < Y, X = Y, or X > Y. We do so in a simple way, compute X − Y and look at the result. If the result is negative, then X < Y; if the result is 0, then X = Y; if the result is positive, then X > Y. Comparisons are used in our programming logic, for instance, in if-then statements like if(x < y) z = x; else z = y. A summary of the circuits found in ALUs is provided in Table 2.1. In this table, we reference status flags.

TABLE 2.1 ALU Circuits

Component	Role	Comments
Adder	Performs addition	Often combined as an adder/subtractor; may have two types: integer and floating-point
Subtractor	Performs subtraction	
Incrementor/ Decrementor	Add 1/subtract 1	May use the adder/subtractor or may be a separate circuit
Negation	Negate a value	May also be called a *complementor*; may be a stand-alone circuit or combined with a subtractor
Multiplier	Performs multiplication	Usually three types: unsigned integer, signed integer, and floating-point
Divider	Performs division	
Logical shifter	Shifts bits	May be a single shifter for both left and right shift or may be two separate shifters
Arithmetic shifter	Shifts bits but retains the sign	This shift ensures a negative number remains negative and a positive number remains positive
Rotator	Rotates bits	Like the shifters, may have separate left and right rotates, or there may be a single one
Rotator with carry	Rotates bits but includes the carry flag	Used so that the carry bit can be used as part of a number
Comparator	Compares two numbers	Subtracts the second from the first, setting the appropriate status flag bit (PF, ZF, NF)

Abbreviations: ALU, arithmetic-logic unit; NF, negative flag; PF, positive flag; ZF, zero flag.

This collection of circuits operates on electrical current sent from the register file. Current is in one of two forms, high voltage or no (or low) voltage. This allows us to represent 1s and 0s. We store our binary values as current or no current both in registers and memory. These values then are moved from one location of the computer to another over the bus. In Figure 2.5, we see the register file connects to two locations in the ALU; these are the two inputs such as X and Y, and the ALU connects back to the register file. Additionally, lines from the control unit select which of the circuits should perform the operation on the two data. For instance, if the operation is an ADD, then the control unit selects the adder. The result comes out of the adder and flows right back into a register in the register file.

All ALU operations produce results other than the numeric value from the specific operation. These extra results are of interest to the processor because they inform the processor of choices it can make. These results are stored in a special register called the *status flags*. Each bit of this register represents a different type of result. For instance, when performing an addition, the result could be positive, negative, or zero and may cause a carry out from the leftmost bit, possibly resulting in an overflow. An overflow is an error because the result of the addition is too large to place back into a register. Table 2.2 lists some of the more common status flags found in processors and how the CPU will use that result.

2.2.3 The Control Unit

The *control unit* is the circuitry in the CPU that manages the fetch-execute cycle. The control unit steps through each part of the cycle, sending out commands in the form of signals across the bus. These signals are sent both to components within the CPU (e.g., selecting a circuit in the ALU or registers in the register file) and components outside of the CPU (such as a memory read signal or an output signal).

To perform the fetch-execute cycle, the control unit needs several pieces of information:

- Location of the next instruction
- Current instruction
- Result of previous instruction

TABLE 2.2 Status Flag Bits

Flag	Meaning	Use
Positive	Result was a positive number	Used when condition tests >
Zero	Result was 0	Used when condition tests =
Negative	Result was a negative number	Used when condition tests <
Carry	Result produced a carry out	Used to obtain an extra bit of storage than what can be stored in the data register
Overflow	Result was too large to be stored	Indicates an error should be raised
Parity	Indicates even or odd parity	Used for error handling; we explore this in Chapter 4
Interrupt	Indicates a device wishes CPU attention	Used so that the CPU can focus on a device or another process; we cover interrupts in Chapters 3 and 5
Mode	Indicates operating mode	User mode or privileged mode indicates whether all of memory and all of I/O can be accessed or not; we look at the two modes in Chapter 5

Abbreviations: CPU, central processing unit; I/O, input/output.